Self-Organising in Blockchain Infrastructures: Generativity through Shifting Objectives and Forking

Abstract

Given the ubiquity of digital technologies, and increased use of autonomous algorithms, it is likely that many of today's social and organisational processes will one day include autonomous elements. The Bitcoin blockchain is likely the first case of an increasingly generative and autonomous way of organising—and one in which the blockchain itself, as a digital infrastructure, plays a crucial role in how, and which, organising objectives are realised. The specific properties of blockchain infrastructures—distribution of control, openness to manipulation, and generativity of the underlying source code—make it an ideal case to study patterns of what we call 'self-organising'.

This paper investigates the phenomenon of self-organising through a study of forking in the Bitcoin blockchain infrastructure between 2010 and 2016. It adds to the emerging body of research on digital infrastructures, and particularly blockchain infrastructures, by conceptualising forking as a pattern of self-organising in blockchain infrastructures, specifically involving: 1) the underlying infrastructure; 2) the scale of code changes; 3) individual objectives; and 4) collective adoption, whether specific or general. Thus, it shows how forking in blockchain infrastructures mediates between divergent organising objectives and existing capabilities, on the one hand, and generates self-organising on the other hand. It further contextualises these findings in extant work on digital infrastructures, offers a guide for designers of blockchain infrastructures and proposes the concept of 'generative mirroring' as a pattern wherein blockchain infrastructures and organising adaptively co-evolve.

Keywords: Bitcoin, Blockchain, Digital infrastructure, Forking, Self-Organising, Generativity

Introduction

One of the most anticipated applications of blockchain technologies (hereafter just 'the blockchain') is that of the autonomous, self-governing and self-regulating infrastructure, for instance through a Distributed Autonomous Organisation (DAO) ("The Bitcoin Hype And The Potential Disruptive Power Of Blockchain Technology" 2018; Beck et al. 2016; Chapron 2017; Peters and Panayi 2015; Wörner et al. 2016). Once fully realised, these DAOs, the archetypical instance of distributed organising, are expected to be fluid and digital in nature—without offices, managers, contracts, policies or payrolls, and without centralised strategic agendas (Barrett et al. 2016). Instead it is believed that reliable time stamps and automated, verifiable decision processes will allow not just for fully automated organising processes at scale, but for more reliable organising as transparent digital processes replace *ad hoc* social ones (e.g. Markus and Robey 1983).

However, we are a long way from understanding what it takes to facilitate the creation of a robust and transparent DAO: the interplay between the autonomous and distributed capabilities of a digital infrastructure like the blockchain, and the social objectives that can be realised through it, are still poorly understood. Moreover, early DAOs and first generation blockchain infrastructures have been crippled by tensions between the capabilities of the blockchain and the objectives of individuals and communities (e.g. Andersen and Ingram Bogusz 2017; Avital et al. 2016). For instance, an Ethereum-based DAO was brought down in May 2016 (see Leising 2017) due to flaws in the protocol that governed the DAO—which allowed for the theft of tokens worth around 150 million USD at the time. This was partly because the code itself had a loophole that an individual could exploit, and partly because the dynamics around such exploitation are poorly understood. The theft represents the first of many instances of self-organising, where the capabilities of a blockchain infrastructure and heterogeneous objectives intersect and lead to new patterns of organising and to the pursuit of new objectives.[1] Our understanding of this notion of self-organising and the dynamics that drive it are still under development: blockchain governance, decision rights, accountability, and incentives have all been highlighted as sources of potential friction (Beck et al. 2018), but have never to our knowledge been explored in the context of organising. This lack of understanding around the interplay between technology capabilities and organising in the blockchain context may contribute to the observation that radical organisational transformations may take more time than expected (Iansiti and Lakhani 2017) as well as to the failure of some early instances of self-organising mediated by the blockchain.

In this paper, we zoom in on organising at the intersection of digital blockchain infrastructures and distributed social objectives, or what we call 'self-organising'. Here, we define self-organising as involving changes in structures, processes and boundaries between independent actors guided by recursively applied, code-enshrined rules (Drazin and Sandelands 1992; Holland 1992; Pettigrew et al. 2001). We further take into account the role of digital artefacts, in this case the blockchain as a digital infrastructure, in these processes (Leonardi and Barley 2008). We build on the observation that the fulfilment of organising objectives through the blockchain entails distributed organising, leading to unclear chains of cause and effect (Beck et al. 2018). Indeed, the dynamics between a distributed blockchain infrastructure and distributed objectives, and how they lead to the fulfilment of diverse objectives—more broadly thought of as organising—remain to be explored.

Extant research has shown that the most efficient pursuit of organising objectives can be facilitated by organising that 'mirrors' the underlying digital artefact(s) upon which it depends

[1] Although the DAO case is an interesting illustration, as an isolated incident it lacks the detail and dynamism needed for a deeper study of self-organising.

(Burton and Galvin 2018; Tiwana et al. 2010). While the objectives themselves are believed to evolve in response to exogenous environmental changes (Henfridsson and Bygstad 2013), artefacts are treated as controlled by the organisation itself, with little influence over the resulting organisational objectives. Instead, hierarchical organisations have clear hierarchies and *loci* of control: aligning digital capabilities and organising objectives through mirroring is therefore a question of top-down strategy and design rather than co-ordination (Lee and Berente 2012). However, a digital infrastructure, defined as a "shared, unbounded, heterogeneous, open, and evolving sociotechnical system comprising an installed base of diverse information technology capabilities and their user, operations, and design communities" (Tilson et al. 2010: 748), has no clear hierarchy or locus of control. Instead, digital infrastructures are characterised by distributed control, fluid technological capabilities, and openness to manipulation by a large and heterogeneous community. Top-down governance is therefore not only unpopular, it is often impossible (O'Mahony and Ferraro 2007). If there is going to be alignment between capabilities and the pursuit of objectives—and there is no guarantee that there will be—this will need to be emergent and cannot be controlled in the same way as in a hierarchical organisation. Thus, the possibility of pursuing certain objectives, and thus organising more generally, likely cannot be controlled *ex ante*.

Three characteristics specific to blockchain infrastructures affect whether—and how—diverging organising objectives can be realised: 1) distributed control of decision rights through transaction verification (Beck et al. 2018); 2) openness to manipulation of infrastructure capabilities (Campbell-Verduyn 2018); and 3) generativity of organising objectives (Andersen and Ingram Bogusz 2017; Henfridsson and Bygstad 2013). First, algorithms called consensus mechanisms set the recursively-applied rules that govern which node verifies transactions in the distributed ledger (Bano et al. 2017; Chiravuri et al. 2011). As a result, the addition of new transactions to the ledger is controlled by a network of distributed verifier nodes. Second, the source code dictates how the infrastructure operates, including the rules whereby platforms, applications and other modules connect to it (Nyman and Lindman 2013). For most public blockchain infrastructures, this code is open to manipulation by an Open Source (OS) community (Howison and Crowston 2014), meaning that some of what we know about OS communities may be relevant to understanding the blockchain. Lastly, although code and algorithms play a key role, so too do the shared objectives of the members of its distributed community of users, developers, and miners (Beck et al. 2016).

We observe that these three characteristics of blockchain infrastructures, and with them technological and social processes, are not severable: organising does not originate from either the technological capabilities *or* the individuals involved. Instead, it emerges as 'self-organising' interactions between independent actors in a distributed system, governed by recursively applied (code-based) rules that generate stable social structures (Drazin and Sandelands 1992; Holland 1992). Thus, self-organising entails patterns and regularity emerging without central control, whether individual, organisational or digital (Anderson 2012).

Research into the relationship between organising objectives and the evolution of blockchain infrastructures falls short when it comes to understanding self-organising in the face of the complex interdependencies described above. In this paper, we therefore ask the following question:

> How does self-organising occur in blockchain infrastructures?

We address this research question through a multi-method, longitudinal case study of self-organising in the Bitcoin blockchain infrastructure over the course of six years. The Bitcoin blockchain, while a digital infrastructure, does not have an organisational legacy (see, e.g., Star and Ruhleder 1996). Instead, the infrastructure was created in isolation: by an anonymous individual or group of individuals who later severed ties with the project. As such, Bitcoin does not have an *ex ante* organising base (hierarchical or otherwise), as it was severed from whatever organising base it once had. Instead, self-organising emerged from and around the infrastructure as it evolved. It can therefore be distinguished from both hierarchical and digital infrastructures more broadly, as discussed further below.

The case of the Bitcoin blockchain is instructive in providing insight into the larger phenomenon of self-organising (Anderson 2012), generativity associated with blockchain infrastructures (Siggelkow 2007; Wörner et al. 2016), and ultimately organising as a more autonomous process mediated by technological capabilities. Moreover, as it is generally considered the parent of subsequent generations of blockchain infrastructures (Avital et al. 2016), its ability to generate self-organising is likely to help us understand the pursuit of diverging objectives in second (and third) generation(s) of blockchain infrastructures (e.g. Ethereum, Hyperledger, etc.). However, for the sake of being able to draw clear empirical boundaries we focus only on the Bitcoin blockchain infrastructure in this paper.

For simplicity, we will refer to the emergent social activities around the Bitcoin infrastructure as the 'community', the heterogeneous objectives that emerge pursuant to organising as 'organising objectives', distributed ledger technology in general as 'blockchain', and to a specific record of transactions as the 'distributed ledger' or the 'ledger'.

This paper is structured as follows. First, we examine the existing literature on digital infrastructures to identify known organising principles and outline the differences and similarities between digital and blockchain infrastructures. Based on our review of existing literature, we substantiate the notion of self-organising in blockchain infrastructures before proposing and elaborating on how code changes, and code forking specifically, act as mechanisms of self-organising. Thereafter we present the case background and research design, before showing how code forking in the blockchain infrastructure mediated organising shifts in the Bitcoin community. Finally, we discuss our research findings and their implications for self–organising mediated by blockchain infrastructures.

Self-Organising in Blockchain Infrastructures

In November 2018, the Github account for Bitcoin Core, the first generation blockchain, suggested that its source code has been copied for re-use over 21 000 times, that 18 500 changes had been made to the original source code,[2] and between June 2013 to November 2018 the number of people following its development had risen from 2 300 to over 35 000.[3] This combination of high levels of activity, a growing user base, and divergent goals and objectives points to the idea that the blockchain is not only a project with a significant following, but also one that is constantly adapting, leading not only to improvements in the code, but also to new permutations with new commercial and non-commercial objectives. These shifts, in turn, lead to self-organising in its community of users, developers, and miners.

In the following we first review existing theoretical views on the interplay between infrastructures and organising before highlighting the specific characteristics of blockchain infrastructures that enable self-organising. Against this backdrop, we then conceptualise how self-organising occurs in, and in perpetuation of, blockchain infrastructures.

---

2 https://github.com/bitcoin/bitcoin/network/members

3 https://www.timqian.com/star-history/#bitcoin/bitcoin

Existing Views on Infrastructure Organising

Infrastructures have been argued to possess an "…overall capacity to produce unprompted change driven by large, varied, and uncoordinated audiences" (Zittrain 2006: 1980), or what is called generativity. The result is emergent (Edwards et al. 2007) and even 'accidental' progress, or innovation (Austin et al. 2011) through slow and incremental changes. However, previous studies have not explored the patterns that explain how digital infrastructures, apart from those wholly controlled by single organisations (Ciborra 2000; Eaton et al. 2015), generate this progression. This begs the question of whether infrastructures, in an age of increased automation and digital ubiquity, participate in their own evolution, and to what extent. Understanding how digital infrastructures perpetuate their own existence helps us unpack the relationship between digital infrastructures and social objectives during the process of (self-)organising.

Received literature on digital infrastructures represents at least four distinct views of how social practices, some of them hierarchical, give rise to digital infrastructure evolution; namely through adaptation, inscription, interaction, and managerial choice. These views are rooted in different bodies of literature.

First, *adaptation* views see infrastructure evolution as a result of the efforts of distributed human actors to adapt to their environment and to other actors (Braa et al. 2007; Hanseth et al. 2006). Adaptation views build on developments of complexity theory (Holland 1995; Mol and Law 2002). For example, Hanseth & Lyytinen (2010) argue that digital infrastructure design can account for the adaptability of distributed actors, and Nan (2011) explores how the use of distributed digital technology emerges as adaptations between users, technology and tasks.

Second, *inscription* views describe how infrastructures evolve as human actors translate and inscribe their interests into assemblages of technological components. Building on actor network theory (Callon 1986; Latour 1987), inscription views emphasise the relationship between human actors and technology in translating and inscribing behaviour in structural terms. They see infrastructure evolution as changes to a set of relations between humans and technology as human actors mobilise resources to support some organised action (Aanestad and Jensen 2011; Hanseth and Monteiro 1997; Yoo et al. 2005). For example, Eaton et al. (2015) describe how the tuning of boundary resources by a network of distributed human actors affects digital infrastructure evolution.

Third, *interaction* views argue that an infrastructure's evolution entails continuous interaction between its users and stakeholders as they engage in sensemaking around an organised activity. Drawing on theories of collective learning and work practices (Lave and Wenger 1991; Wenger 2000), interaction views see infrastructure evolution as a result of interactions within a community-of-practice resulting in the formation of socio-technical relations (Pipek and Wulf 2009; Star and Ruhleder 1996; Vaast and Walsham 2009). For example, situated learning by OS community participants leads to the ability to advise others, as well as the ability to improve the code (Fang and Neufeld 2009).

Finally, *managerial choice* views emphasise the role of management decisions in facilitating infrastructure evolution. Infrastructure evolution is seen as a process by which managers initiate and implement changes to information technology infrastructure in order to align strategic capabilities and strategic objectives (Beckert 1999; Child 1997). For instance, Broadbent & Weill (1997) explain how managers—based on their understanding of the strategic context of their organisation—determine the infrastructure capabilities they should implement to achieve their business objectives. The notion of 'mirroring', although rooted in a different intellectual tradition, is consistent with this view (Burton and Galvin 2018).

Table 1. Organising Principles in Digital infrastructures			
Organising principle	**Description**	**Theoretical foundation**	**Example references**
Adaptation	Distributed actors adapt to their environment through changes in tasks, technology and relations	Complexity theory	Hanseth & Lyytinen (2010) Nan (2011)
Inscription	Existing organising practices are inscribed in technological artefacts	Actor Network Theory	Aanestad & Jensen (2011) Eaton et al. (2015) Yoo et al. (2005)
Interaction	Interactions in a community of practice resulting in new socio-technical relations	Collective learning and communities-of-practice	Fang & Neufeld (2009) Pipek & Wulf (2009)
Managerial Choice	Choice of infrastructure governance and organising as a result of informed management decisions	Strategic choice theory	Beckert (1999) Broadbent & Weill (1997) Child (1997)

The views described above and outlined in Table 1 all share three main assumptions about infrastructure organising: first, that infrastructures are built upon, and follow from, pre-existing organised practices (Star and Ruhleder 1996). Second, in the course of infrastructure evolution, human behaviour is inscribed onto the technological components of the infrastructure (Hanseth and Monteiro 1997), and third that changes to the infrastructure require coordination among heterogeneous and distributed human actors (Ciborra 2000; Hanseth and Lyytinen 2010). In other words: human objectives and practices, including coordination, determine digital infrastructure evolution.

However, the applicability of these assumptions to the Bitcoin blockchain is unclear, and has not been tested in previous blockchain studies. In particular, the Bitcoin blockchain has few, if any, pre-existing organised practices; the centrality of its recursive, code-based rules mean that it is not controlled exclusively by human objectives and practices, and changes to the infrastructure can occur even without coordinated activities.

Our aim is not to validate these assumptions here, but rather use what we already know about digital infrastructures as a departure point for developing an understanding of how the infrastructure itself, through code-based consensus mechanisms and adoption, leads to emergent self-organising. In the following, we define and distinguish blockchain infrastructures before proposing and substantiating a specific organising mechanism: the forking of infrastructure capabilities pursuant to organising objectives.

Organising in Blockchain Infrastructures

Blockchain has much in common with other infrastructures: control is not centralised, but rather dynamically negotiated (Weill and Broadbent, 1998), and it is comprised of a layer of code upon which both platforms and applications are built and upon which organising practices rely (Tilson et al. 2010; Vaast and Walsham 2009). As such, it is not just an OS project maintained by a decentralised community, but rather a specific sub-type of digital infrastructure, which supports a set of unique and distributed relational practices through which it is organised (Star 1999). Implementations of public blockchain infrastructures (Beck et al. 2018), in particular, are best described as a new breed of digital infrastructure (Bygstad 2010; Hanseth and Lyytinen 2010).

Although a blockchain infrastructure belongs to a class of digital infrastructure in that it is a “shared, unbounded, heterogeneous, open, and evolving sociotechnical system comprising an installed base of diverse information technology capabilities and their user, operations, and design communities" (Tilson et al. 2010: 748), the constraining properties of the blockchain mean that they likely operate differently to infrastructures that have previously been studied (e.g. Iannacci 2010; Kuk and Janssen 2013). Extant research has instead examined how *ex ante* organising structures (Henfridsson and Bygstad 2013; Star 1999) and boundary objects (Eaton et al. 2015) determine infrastructure organising and evolution, while acknowledging that control is often dynamically negotiated and not necessarily fully centralised (Weill and Broadbent, 1998).

There are, further, at least three reasons why blockchain infrastructures behave in ways not previously discussed in received literature on digital infrastructure organising. It is beyond the scope of this paper to explain all of the technicalities of blockchain technology (but see e.g. Böhme et al. 2015). However, three constraining properties are especially relevant in understanding self-organising in blockchain infrastructures: distributed control (Beck et al. 2018), openness to manipulation of technological capabilities (Avital et al. 2016), and the generativity of organising objectives (Hanseth and Lyytinen 2010; Nærland et al. 2017).

First, blockchains are not just decentralised, but distributed: they consist of multiple instances of a ledger that are shared and maintained between a disparate group of actors linked by shared code and without any hierarchical social structure (Bano et al. 2017; Brenig et al. 2016; Guerini and Moneta 2017). Also, they are open in the sense that their technological capabilities are open for manipulation. In the case of public blockchain infrastructures such as Bitcoin, a community of distributed enthusiasts are behind these manipulations (Beck et al. 2018; Campbell-Verduyn 2018). Many of these individuals are also members of a larger OS community used to tweaking code at will—and without legal or organisational impediment (Fang and Neufeld 2009). This openness to manipulation leads to increasing heterogeneity of both how the underlying infrastructure is used, and the resulting organising objectives (Ingram Bogusz and Morisse 2018). Put together, these characteristics mean that a new technological capability, through code, can be proposed and developed by anyone. Thus, the dynamics surrounding code development and adoption help us understand how dynamic self-organising around an emergent organising objective emerges. However, community adoption plays a role in determining whether new (sets of) capabilities are discarded or adopted *en masse*. Further, the source code's characteristics and the features of proposed changes to the code affect how important adoption is for the evolution of the infrastructure. Indeed, dynamics between the source code and proposed changes, mediated by adoption requirements, determine whether manipulations are incorporated into the infrastructure at all. As such, the capabilities of new as well as pre-existing code have implications for how the infrastructure operates in the future. These capabilities influence organising activities mediated by the infrastructure, including self-organising, by enabling and limiting what members of the community can do.

For instance, the distributed process whereby a new transaction is recorded ensures ledger immutability: a transaction is transmitted by one member of the community to others in the network for them to verify that it is legitimate and consistent with previous entries (e.g. so that it does not include already-spent tokens or come from a fake account), as well as to record it. The software is designed so that transactions can only be added onto the ledger after verification, and cannot be changed or removed once entered without changes to the entire blockchain infrastructure.[4] Should conflicting versions come to exist, the blockchain protocol stipulates that the version of the software, which includes the ledger, held by the majority of those doing the verification ('miners') is the 'real' blockchain (Nakamoto 2008; Taylor 2013). This version control system not

---

[4] Although there is some discussion around how much control is required to retrospectively change the blockchain, see e.g. Eyal, I. and Sirer, E.G., 2014, March. Majority is not enough: Bitcoin mining is vulnerable. In *International Conference on Financial Cryptography and Data Security* (pp. 436-454). Springer Berlin Heidelberg.

only makes the ledger itself immutable, it makes the protocol that runs it subject to the same majority-rule requirement. Individual actors therefore cannot change either the ledger or the protocol. Instead, if one is unhappy with the capabilities of the current implementation, the OS nature of the code means that even fundamental infrastructure capabilities are open to manipulation (Nyman 2015).

Distributed control and openness to manipulation of technological capabilities has another effect: for fundamental changes, known as forks, to be made to the blockchain, the majority of miners have to adopt them. When this occurs, those miners running the version that is in the minority are seen to be running a *de facto* alternative. That is, they are no longer running a compatible version of the infrastructure: neither the source code that they run nor the transactions entered into minority-held alternative ledger can be recognised by the original blockchain. This is, however, only true when the versions are inconsistent with one another; more subtle rules apply when minor updates of the code or consistent code additions are involved.

The generation and adoption of new technological capabilities is thus controlled both by the rules and the content of the pre-existing code, and by a distributed community of miners, developers, and users (Catalini and Gans 2016). Blockchain infrastructures thus differ substantially from both the hierarchical and digital infrastructures previously studied, as outlined above and summarised in Table 2. While understanding how they self-organise builds on existing understandings of digital infrastructures, it also allows us to better understand next generation applications of blockchain and related distributed and autonomous technologies.

Table 2. A Comparison of Hierarchical, Digital, and Blockchain Infrastructures

Technical and Social Characteristic(s)	**Hierarchical Infrastructure**	**Digital Infrastructure**	**Blockchain Infrastructure**
Technological capabilities	Clear base of technological capabilities (e.g. enterprise systems)	Fluid technological capabilities (e.g. digital platforms)	Fluid technological capabilities, open to manipulation (e.g. digital ecosystems)
Membership	Clear boundaries, typically drawn along firm boundaries	Large and heterogeneous community (fluid boundaries)	Large, heterogeneous and pseudonymous community (fluid boundaries)
Mode of control	Centralised control (Control by a hierarchical organisation)	Decentralised control (Control relegated across organisational boundaries)	Distributed control (Control through network of distributed verifier nodes)
Instantiation of control	Coordination by organisation members	Designed-in control mechanisms (e.g. through boundary resources or control points)	Consensus mechanisms contained in code
Organising objectives	Defined top-down	Interdependent, but often centralised objectives	Complex interdependencies, generative objectives

We now move on to explore self-organising in a blockchain infrastructure and how it is actualised through code forking.

Forking as Self-Organising in Blockchain Infrastructures

Given the fact that the evolution of blockchain infrastructure capabilities—and therefore organising—is distributed and emergent, what happens when there is a disagreement about the future of the infrastructure, or what capabilities it should have or enable, from within the community? In a non-distributed or proprietary setting ('hierarchical infrastructure'), the disagreement would be solved with reference to the organisation's hierarchy (Burton and Galvin 2018; Dahlander and Magnusson 2005; Kartseva et al. 2010). However, among both digital and blockchain infrastructures, the answer is less simple: in general, members of the community try to come to a negotiated settlement (Nyman 2015). In addition to this, blockchain infrastructures rely not just on the protocol at a set point, but also on the historical ledger generated over time. On a practical level, it means that compatibility is determined both based on what the protocol looks like at the time of inquiry, and whether a proposed change is backwards- or forwards-compatible with the existing blockchain infrastructure. This temporality means that a disagreement so fundamental that it splits the community may lead to the emergence of two (or more) inconsistent versions of a blockchain infrastructure.

Previous research into hierarchical organisations has suggested that alignment between technologies' capabilities and organising objectives not only allows for the effective pursuit of changing organising objectives (e.g. Burton and Galvin 2018; Marwaha and Willmot 2006; Tiwana et al. 2010), but that sustained effectiveness requires that technologies evolve in line with changing organising objectives (Hirschheim and Sabherwal 2001). Thus, within established organisations, researchers have examined not just the phenomenon of artefact-organisation alignment, but also pointed to the importance of adaptive evolution of this alignment in line with evolving infrastructure capabilities and organisational objectives. This research has suggested that 1) digital artefact modularity, or the decomposability of a system into components with limited interdependence (Simon 1962), and 2) a loosely-coupled organisational structure (Tiwana and Konsynski 2010), facilitate this adaptive alignment. Indeed, modularity and loose coupling give an organisation both the flexibility to add and subtract functionality on a technical level, and the agility to do so with limited intervention from the rest of the organisation (e.g. Hagel 2002, Tiwana and Konsynski 2010).

What results is what has been called 'mirroring', where the organising objectives that govern an infrastructure reflect its underlying technology (Colfer 2007; Colfer and Baldwin 2016). Such mirroring has been identified for instance in areas around organisational control (e.g., Weill and Ross 2005; Tiwana and Konsynski 2010), as well as pursuant to product development and delivery (e.g. Colfer and Baldwin 2016; MacDuffie 2013). Mirroring has been argued to occur for reasons around effectiveness (Burton and Galvin 2018). Arguing from the perspective of transaction cost economics, Burton and Galvin (2018) point to how hierarchies are superior to decentralised markets because they allow for more effective pursuit of goals (see also Geyskens, Steenkamp, and Kumar 2006), and reduce technological uncertainty and information overload (Weber and Mayer 2014, 345). However, extant research has pointed to how high levels of infrastructural complexity might make it more effective to 'mist' the mirror: making alignment and adaptive alignment more limited (Burton and Galvin 2018). This observation has been echoed in studies of OS committal structures: when tasks are routine and risks are low, centralised control is thought to be the most effective form of governance (AlMarzouq et al. 2015). However, centralisation and hierarchies are thought to have a chilling effect on innovation (Gawer and Cusumano 2014), and to be less effective forms of management when tasks are complex and high-risk (AlMarzouq et al. 2015). Thus, when an infrastructure is complex, emergent and not centrally controlled, as with a blockchain infrastructure, not only is alignment with formal organisational objectives tricky, it is likely to be ineffective.

An increasing number of digital infrastructures are neither developed nor maintained by hierarchical organisations, most notably those developed and governed by OS communities (O'Mahony and Ferraro 2007) and in innovation ecosystems (Wareham et al. 2014). Blockchain infrastructures are a relatively recent addition to this list. Although transaction cost economics would suggest that such distributed organisations are ineffective and characterised by high levels of uncertainty, research has shown that non-centralised organisations not only develop new modes of governance and control (O'Mahony and Ferraro 2007), but also deal with uncertainty and information overload in different ways to hierarchical organisations (Jones et al. 2004). In the case of the blockchain infrastructure, we propose that one way in which both control and effectiveness are maintained in the face of complexity and temporal dynamism is through the 'generative mirroring' of infrastructure capabilities and organising objectives, in this case through blockchain forking.

Forking of infrastructure capabilities takes place when source code is copied and modified in spin-outs known as 'forks'. Forking is defined as when "a part of a development community (or a third party not related to the project) starts a completely independent line of development based on the source code basis of the project" (Robles and González-Barahona 2012, 137). Code forks are typically frowned upon because they negatively affect community and individual developers' reputations (Nyman 2015; Weber 2004), and because multiple, incompatible versions of a software can discourage related future development (Meeker 2008; Nyman 2015). Nevertheless, both within and beyond existing communities, forks in the source code are a visible reflection of changes in organising objectives. As such, the notion of code forking—in combination with mirroring of infrastructure capabilities and organising objectives—provides a theoretical lens through which digital infrastructure evolution can be studied.

We distinguish three kinds of forks; the *pseudo-fork,* which repurposes existing source code and therefore has no compatibility issues, *development forks*, which are forward-incompatible insofar as they build capabilities that add to the code in the existing ledger, and the *hard fork*, which creates a fork that is either forward- or both forward- and backward-incompatible (Fang and Neufeld 2009; Raymond 1999).

The first two types, pseudo- and development forks, involve the distribution of the original code along new channels. The resulting new distributions of the code are compatible with the old version (backward-compatible) and benefit from future developments in the parent code (forward-compatible) (Nyman 2015). In the case of pseudo-forks, existing source code is repurposed and the fork allows for the application of existing capabilities to achieving new organising objectives. This allows for 'variation' in organising objectives without challenging or requiring changes to existing infrastructure capabilities. Development forks, by adding new capabilities, or new code, to the existing code base while maintaining backward-compatibility of the ledger, 'adapt' infrastructure capabilities to serve new organising objectives. In contrast, hard forks entail a fundamental change to the underlying code such that the new and the old versions of the blockchain are both forward- and backward-incompatible. While other forks vary or adapt existing organising objectives, hard forks support and reproduce radically new objectives that are incompatible with existing capabilities. The radical differences between them may extend to the creation of whole new communities, with new norms and objectives, or separate infrastructures within the same community. As a mode of self-organising, we therefore refer to hard forking as leading to 'speciation' or 'bifurcation' (Andersen and Ingram Bogusz 2017).

In summary, both the forward and backward compatibility of changes to source code are potentially at issue when forks to a blockchain infrastructure occur. These determine the extent of the adoption required for a fork to actualise changes in organising, together with both the capabilities of the original source code and the characteristics of any proposed fork. This is because a blockchain infrastructure contains an immutable record with temporal reference points.

What follows is a description of how we unpacked forking as self-organising through a longitudinal multi-method research design.

Research Design

In order to answer the research question of how blockchain infrastructures evolve new forms of self-organising, we conducted a longitudinal multi-method study (Venkatesh et al. 2013) of generativity in the Bitcoin blockchain infrastructure over a period of six years. In the following section, we first discuss our case selection and its background before detailing our data collection and analysis methods.

Our choice of case was driven by the need to meet four basic requirements: first, the infrastructure had to be predominantly self-organising in the sense that the organisation around it would have emerged in tandem with identifiable technological changes. Second, we had to be able to identify distinct instances of forking. Third, the case setting needed a history spanning over a longer period of time allowing us to study it in both detail and at scale. Finally, the infrastructure needed good records of both code forking and organising practices to allow us to analyse the dynamics of code forking and self-organising.

The Bitcoin blockchain is self-organising, relatively long lived, and has good digital trace records, and therefore fulfils these requirements. Although the OS software has been re-used to create new infrastructures (e.g. Ethereum, Ripple), these next generation applications are still emergent and, as cases, are still nascent and therefore trickier to study (Yin 2003).

Data Collection and Analysis

Our aim with this research was to identify instances of forking as self-organising, as well as to tease out their antecedents and implications, in order to understand self-organising in blockchain infrastructures more generally. To this end, we needed to establish a sequence of forking events, and explicate both the patterns that generated them, and their consequences. Our data collection and analysis therefore took into account these objectives.

Overall, our methods were grounded in inductive reasoning and rested on the use of two sources of data: digital trace data from the Bitcoin online community, and extensive documentation (see Table 3). This was in order to 1) imbue our computational findings with context, as provided by documents (Gaskin et al. 2014), and 2) to ensure the veracity of our findings.

Our primary data source was forum data, which we supplemented with documentary data. Although it is commonplace to study code repositories when examining code-level activities (e.g. Dabbish et al. 2012), a weakness of this approach is that it does not capture either the context in which code-level activities (forks) occur, or their implications. In contrast, forum data provides good records of both when forks occurred (or even potentially could have occurred) and the underlying social context (Gaskin et al. 2014). Forum data was therefore chosen over code repositories as our primary data source because our main focus was to analyse a specific phenomenon, namely self-organising in blockchain infrastructures (in line with Grisot et al. 2014), rather than to follow an OS development project through a code repository (Howison and Crowston 2014; Lindberg et al. 2016). Instead, the fine-grained semantic data contained in the forums allowed us to interrogate the motivations and context in which code-level change occurred, as well as the corresponding changes in organising objectives and practices. Additional documentary data sources were used to mitigate the risk of losing context when conducting computational analyses (Gaskin et al. 2014) and to distinguish proposed forking events from actual forking events in the Bitcoin community.

Table 3. Overview of Data Collection and Analysis			
Data Source	**Description**	**Analytical Techniques**	**Research Outcome**
Digital traces	314 551 digital trace records of interactions collected from the bitcointalk.org community over a six-year period, including records referring to the Bitcoin source code as well as organising changes	Latent Dirichlet Allocation (LDA), the results of which were coded to identify the main forking events that occurred, and the context in which they occurred	Identification of forking events and characterisation of changes in organising objectives during each forking event
Documents	56 Press articles 71 Blog posts on topics related to forks and other conflicts (e.g. political ideologies) in the community from other sites (e.g. Bitcoinfoundation.org, Coindesk.com, Techcrunch.com, medium.com)	We used open and axial coding to produce analytical memos (Miles and Huberman 1994) around the significant code forks, understand the background of the community, and highlight major associated events. This enabled us to verify findings in the primary data source, namely the forum data.	Identification of environmental conditions and important demarcations in the history of the Bitcoin community

We collected the digital trace records by scraping community interactions from the forum bitcointalk.org (Hedman et al. 2013; Howison et al. 2011). In total, we collected 314 551 interactions posted between October 2010 and June 2016. Bitcointalk.org is a forum dedicated to discussions around Bitcoin, primarily in English. It is among the most prominent forums used by Bitcoin enthusiasts. However, unlike forums like Reddit.com, it is often used specifically by Bitcoin professionals meaning that interactions on Bitcointalk.org are more closely linked to the development of the Bitcoin code base. Furthermore, it contains sections that are both general and specific in nature; for instance, threads around the technicalities of the Blockchain and mining, as well as discussions of a more organisational nature.

We analysed the forum data (from October 2010 to June 2016) as illustrated in Figure 1. Open coding of the data was conducted using the computational natural language processing technique Latent Dirichlet Allocation (LDA) implemented in the open source statistical software R (Blei et al. 2003). This process of applying computational techniques to digital trace data is not new to Information Systems research (see, e.g., Berente et al. 2018; Hedman et al. 2013; Howison et al. 2011). LDA is a generative topic model that reveals patterns in a set of documents by extracting unobserved groupings (latent themes) based on semantic similarities between different parts of the data (Sievert and Shirley 2014). LDA discovers latent themes within a collection of documents by sampling a topic for each word at every iteration of the algorithm and ranking words based on their relevance to each topic, which therefore has a unique distribution over words that can be compared using cosine similarity measures (Chuang et al. 2012). This led to the identification of semantic topics for each time interval, analogous to open codes in manual coding (e.g. Glaser and Strauss 1999). These initial open codes were then organised by the algorithm into semantic clusters determined by co-occurrence of terms.

We then manually coded the resulting semantic data clusters to identify the topic of each 'conversation'. In many ways, this resembled the axial coding conducted in qualitative studies (Vaast and Walsham 2011). The process focused on identifying originating and emerging objectives, which we compared against analytical memos generated during the coding process, and in light of our own knowledge of the community and the 127 additional documents to which we referred (Miles and Huberman 1994). This comparison allowed us to relate each shift in objective to a specific forking event, as well as identify what occurred before and after each forking event. Figure 1 describes the process that gave rise to Table 4, discussed in our findings, which provides further detail around the originating and emerging objectives, their link to specific forks, and the identified forking types.

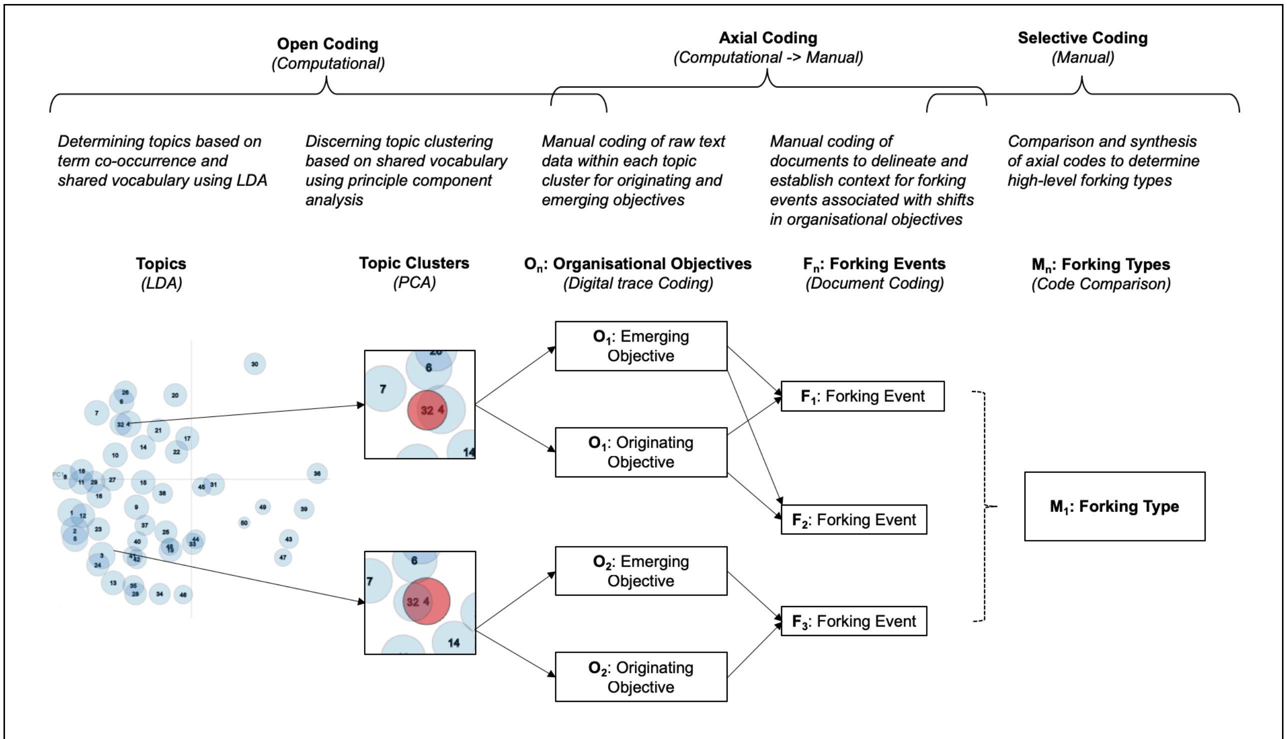

Figure 1. Overview of Data Analysis

Thus, while forums provided the primary data (and supported the main analytical findings), analytical memos based on additional data provided vital context. The analysis combined computational and manual coding and analytical techniques to generate a longitudinal analysis of the organising objectives related to specific code forking events over six years. Based on this analysis, we produced a thick description (Bechky 2006) of code forking as self-organising in the Bitcoin community, which allowed us to explicate self-organising in blockchain infrastructures as a mode of organising more generally.

Having discussed our methods and the background to our study, we turn now to presenting our findings, before discussing their theoretical and practical implications.

Findings: Self-Organising in the Bitcoin Community

Our longitudinal analysis of bitcointalk.org revealed three distinct patterns by which forking of the underlying infrastructure mediated the pursuit of new emerging organising objectives, and the resulting self-organisation through adoption. Although self-organising originated in shifts in objectives articulated at the level of code, adoption defined how—and to what extent—changes in technological capabilities supported evolving organising objectives.

What is interesting to note is that forks occurred in response to shifting organising objectives, and thus indirectly in response to changing environmental conditions. This observation is consistent with existing understandings of infrastructure as including organising practices and contexts, as well as artefacts (Star and Ruhleder 1996). As this was a longitudinal study, we could examine not only when and how code forking in the blockchain infrastructure supported new organising objectives, but also when new ideas were incorporated into the existing infrastructure. The three distinct forms of code forking i.e. pseudo-forks, development forks, and hard forks, their environmental conditions, and an overview of the sequence of forking events of importance to the online community are contained in Figure 2.

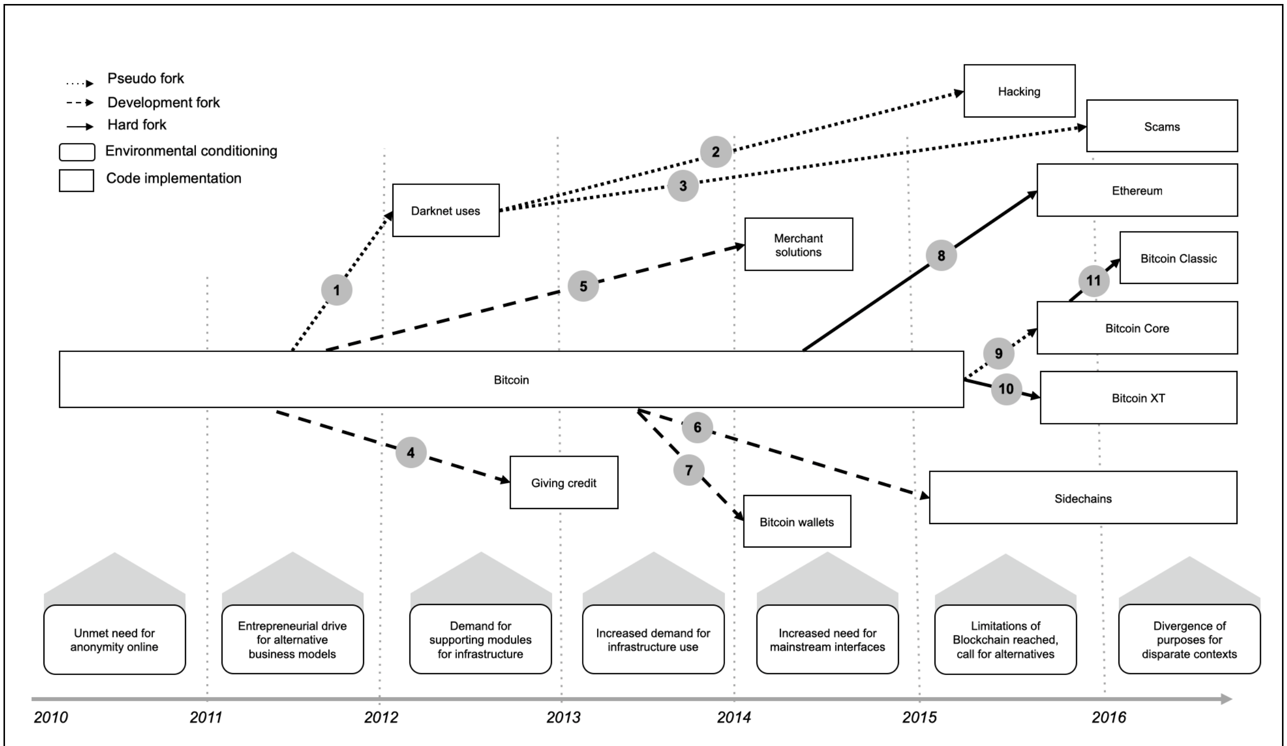

Figure 2. Forking of the Bitcoin Infrastructure

As can be seen in Figure 2, forking in the Bitcoin blockchain infrastructure sparked significant divergence of both source code implementations and organising objectives. Moreover, the occurrence of these patterns correlates with shifts in environmental conditions that encouraged heterogeneous groups to rally around different objectives. Of note is the fact that the underlying code in the blockchain only allowed some of these objectives to be actualised directly (here, pseudo- and development forks). Instead, a hard fork was needed if the objectives being pursued required a fundamental shift in the capabilities of the underlying infrastructure.

Overall, these forks enabled either ‘variations’ on existing objectives (in the case of pseudo-forks), ‘adaptations’ that added complementary capabilities to the blockchain infrastructure (in the case of development forks), or the ‘bifurcation’ of a group in order to pursue objectives inconsistent with what was enabled by the existing version of the infrastructure, sometimes even leading to a new ‘species’ of infrastructure (where a hard fork was required). These shifts, summarised by forking type, are described in Table 4 and elaborated upon further below.

Table 4. Forking Types and Associated Shifts in Organisational Objectives				
Forking Type	**Forking Events (F_n)**	**Originating Objectives (O_n)**	**Emerging Objectives (O_n)**	**Illustrative Quotes**
Pseudo-forking (M_1)	F_1: Darknet F_2: Hacking F_3: Scams	O_1: Providing a decentralised alternative to the financial system	O_1: To illicit purposes for own gain	Usually stuffs on dark net are illegal ones thats why its very risky to make purchases on that place and there are lots of smuggled or even scammers on selling too good to be true offers. April 15, 2016, 01:37:20 PM
	F_9: Bitcoin Core	O9: Response to attempts to make blockchain less distributed through increased block size	O9: To defend existing distributed character of blockchain	We need to realize that split only does harm to bitcoin ecosystem. I know that they acted a bit 'conservatively' but it was a defensive move to protect bitcoin. Although I disagree with his behavior, I am deeply disappointed about how easy [redacted] is willing to split the community in half in excuse of an issue which can be resolved through human talk. August 11, 2015, 09:55:28 PM
Development forking (M_2)	F_4: Credit	O_2: From generating new BTC at algorithmically defined pace	O_2: To adding capabilities of supplying loans and putting up capital for leverage	I`ve invested about 0.1 BTC on Bitbond on an 6W Loan.Iam really exicted gettin it paid BAck. This is my first Investment on Bitbond. I think the rates for the Investors are pretty good and therefore it comes with you if you go high or low. Otherwise, of course its recommended normally to set more smaller Ammounts on differents loans to get it paid back.... October 22, 2015, 10:29:37 AM
	F_5: Merchant apps	O_3: From safe yet technically complex transaction interface	O_3: To adding solutions for using Bitcoin as a payment infrastructure	Yes, we charge a small fee for every customer, which is nothing new to Bitcoin users. We incentive our BlockPay merchants to grant special discounts, cash backs and loyalty rewards for those who pay digital. BlockPay "S" and BlockPay integrated is for free for merchants and can be set up within minutes. You do not need any special hardware, and the software is also for free. We singing up new stores every week right now. July 30, 2016, 08:20:53 AM
	F_6: Sidechains	O_4: From ensuring consistency and decentralisation thus avoiding double spend	O_4: To temporarily mitigating the openness and decentralisation of Bitcoin while staying compatible with fundamental capabilities	If both blockchains "see" each other then side chains are easy. Let's say you want to move value from chain A to chain B. The owner spends bitcoins on chain A from any address (say 1Me) into a well-known unspendable address (let's pretend the address prefix is 1chainBxfer). Nodes on chain "B" are watching chain A, perhaps only as a SPV node to see txouts going to 1chainBxfer. When a suitable transaction is found and sufficiently confirmed, a coinbase txn is allowed on chain B that grants coins to the same addresses as the txins on chain A (1Me)…The real question is how to spend in both directions (2-peg is what people have been saying) when chain A is not aware of chain B? I think that the general consensus is that this is impossible which is why the sidechain idea languished for 2 years. April 11, 2014, 03:53:08 PM
	F_7: Wallets	O_5: From storing BTC on heavily encrypted, highly technical systems	O_5: To making it safe and convenient for individuals to store BTC	There's a gap between secure, convenient, and cheap. If you trust me, I sell paper bitcoin wallets in the mail. That's convenient and cheap, and only insecure to the extent that you're trusting me to not rip you off, otherwise they bulletproof.

				If I were to write this program to run on a gadget I could sell you that printed your addresses on a roll of paper, then it'd be secure and convenient, but not cheap. And of course, compiling these programs is secure and cheap (free), but not convenient. The closer these three points of the triangle come together, the sooner Bitcoin will take off into the stratosphere. August 04, 2011, 04:32:03 AM
Hard forking (M_3)	F_8: Ethereum	O_6: From proof-of-concept for alternative financial infrastructure and system	O_6: To building a new generation platform for blockchain infrastructures	When the grand experiment that is bitcoin began, the anonymous wizard desired to test two parameters- a trustless, decentralized database enjoying security enforced by the austere relentlessness of cryptography and a robust transaction system capable of sending value across the world without intermediaries. Yet the past five years years have painfully demonstrated a third missing feature: a sufficiently powerful Turing-complete scripting language. Up until this point, most innovation in advanced applications such as domain and identity registration, user-issued currencies, smart property, smart contracts, and decentralized exchange has been highly fragmented, and implementing any of these technologies has required creating an entire meta-protocol layer or even a specialized blockchain. January 23, 2014, 11:33:17 AM
	F_{10}: Bitcoin XT F_{11}: Bitcoin Classic	O_7: Refactoring the Bitcoin infrastructure so that it can handle larger transaction volumes (through increased block size)	O_7: To refactoring the Bitcoin infrastructure so that it can handle larger transaction volumes	Objectively superior features are developed elsewhere, they may be integrated into Bitcoin. In fact, many already have been. This is where the problem is... consensus takes time in Bitcoin, and a lot of people with personal agendas are influencing the ultimate consensus. These private technologies will have a centralized organization and they will adapt quickly to the needs of the centralized organization and developers will have little say in the matter. They will just be employed to make the changes forced down on them by these organizations. Bitcoin can adopt many of these improved features, BUT the slow consensus system will hamper their progress in the future. The debate about the block size have been going on for years... How much longer will a major change or a added feature take, to reach consensus? April 04, 2016, 05:50:16 PM

M_1: Pseudo-Forks as Organisational Variation

One of the most compelling patterns of self-organising that emerged from our longitudinal examination was how the flexibility of the blockchain source code not only drove substantial changes in infrastructure capabilities, but also permitted variation in the fulfilment of organising objectives when the environment called for them. These instances of variation are visible in forks F_1, F_2, F_3 and F_9, with the first three relating largely to the use of Bitcoin for illicit purposes, and the last related to the juxtaposition and re-labelling of the existing blockchain as 'Bitcoin Core' in response to a hard fork.[5]

Figure 3 illustrates how the relative importance of pseudo-forking events increased over time within the Bitcoin community. The pseudo-forks F_1, F_2, and F_3 show the same overall pattern: a gradual increase in interest over time, with no sharp breaks or sharp increases in interest over time (but see discussion of F_9 below).

Figure 3 further illustrates the significance of June 2011: at this point, the community started to grow significantly. The number of unique users in the community increased from 164 to 1847 and community activity spiked from 2413 interactions in April to 19 159 interactions by June 2011. A number of key events occurred at around this time: first, WikiLeaks and other prominent organisations began to accept the Bitcoin currency (BTC); second, the '2011 bubble' saw BTC prices soar to over 30 USD, only to burst in the following week when accounts at the prominent Mt.Gox Bitcoin exchange were compromised and BTC with a nominal value of over 8 750 000 USD was lost (Ingram Bogusz and Morisse 2018). The combination of an increase in users, and the publicity around these events introduced heterogeneous and diverse objectives to the Bitcoin community. Moreover, this heterogeneity becomes visible as new users either build, or create the market conditions for new capabilities in the blockchain infrastructure.

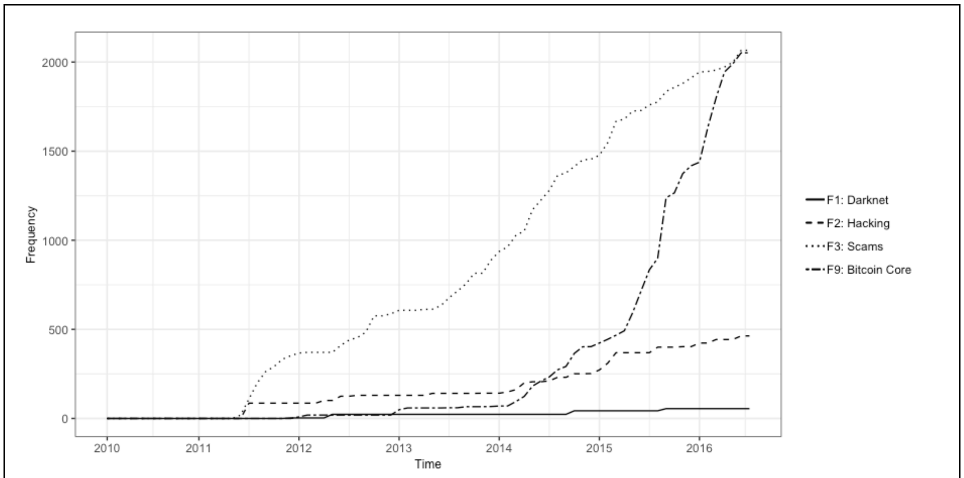

Figure 3. Activity Relating to Pseudo Forks

[5] This relation to a hard fork explains why the corresponding graph in Figure 2 shows steeper growth pattern than the remaining pseudo forks, which are more autonomous

During the early stages of the evolution of Bitcoin in 2011 and 2012, increased heterogeneity in the blockchain community meant that users began to see the new blockchain infrastructure as ripe for conducting transactions outside the established financial system:

…I saw many a lot of devious schemes to earn bitcoin. Just like HYIP, PONZI, Gambling, Bet etc. I am very afraid of RIBA. therefore, for guidance and assistance will be appretiated. Thanks – March 03, 2012, 12:38:21 AM

However, other users were quick to point out how some pseudo forks, notably the ones identified above, broke with norms within the community—and did not represent the objectives of the community as a whole. Nor should these forks be considered synonymous with the technology itself, even though it enabled them:

Of course, bitcoin is not the problem. People who misuse the bitcoin and abuse is are the ones who are the problem. If terorrist are using bitcoin, not really nice of them to shed bitcoin in a bad light. - November 24, 2015, 02:36:19

While 'unpopular' variations dominated, the contextualising of the existing code in response to a split in the community (which also preceded a hard fork) also led to a specific kind of pseudo-fork, namely the defensive re-labelling of existing code.

Defensive Pseudo-Fork

Where changing social objectives were seen as 'legitimate' by the community, and thus widely adopted, what resulted was not a hard fork but a pseudo-fork: the same set of code was infused with new social meaning (here F_9, in May 2015). Our data showed one instance of this occurring, which came in response to a hard fork (F_{10}). It followed similar patterns to a hard fork, namely a sudden rise in interest (see Figure 3), but did not require a change at the level of code to be realised. Instead, defenders of the existing infrastructure labelled the existing infrastructure 'Core', highlighting how its capabilities, and degree of distribution in particular, emerged from a set of 'core' organising goals, notably around democratic control of the infrastructure. Much like the variations in practice represented by the use of the infrastructure for illicit transactions, this kind of fork was an instance of practice-level variation to how the infrastructure was used:

The code which powers the Bitcoin network can be found here: https://github.com/bitcoin/bitcoin. This code has evolved as long as Bitcoin has been around. But the debate over the block size limit and how to manage it has caused some of the best-known developers to set up another client for bitcoin, here: https://github.com/bitcoinxt/bitcoinxt... The former is now referred to as "Bitcoin Core", and the latter "Bitcoin XT". … From the XT github README: "Bitcoin XT is more experimental than Bitcoin Core, and has a strong emphasis on supporting the needs of app developers and merchants. By running it you not only provide additional services to the network but help build confidence in the implementations, contributing towards consensus for inclusion in a future version of Bitcoin Core. - May 31, 2015, 05:28:36 PM

Pseudo-forks, while seemingly something out of the ordinary, did not entail any underlying code change. They simply entailed variation in code use patterns—but nevertheless ones that mediated the pursuit of new objectives, notably ones opposed by other members of the community. Indeed, this opposition—and the freedom to pseudo-fork despite it—was commonplace and was made possible by the nature of the blockchain infrastructure itself. The qualities of distribution and semi-anonymity—seen as elements of building a new technical system based on trust in a system rather than individuals or institutions—paradoxically gave users the possibility to repurpose the system for transactions that many in the community considered to be unethical, including involving drugs on the Darknet (F_1) and Bitcoin-denominated scams (F_3).

Pseudo-forks thus occurred when the existing code facilitated variation in organising objectives. These variations on organising objectives emerged from divergent goals and ideologies, and were manifested through forks to the infrastructure's capabilities. Interpretation, narrative and other events led to the emergence of these variations, while the pre-existing code limited (and enabled) the possible interpretations. Moreover, pseudo-forks attracted additional community members and influenced dynamics between existing members, affecting how the community interacted.

Having described how pseudo-forks enable and constrain self-organising by enabling changes in organising objectives to existing source code, we now look closer at how self-organising manifests when the changes to infrastructure capabilities are more substantial.

M_2: Development Forks as Organisational Adaptation

Variations in organising objectives were not the only forms of self-organising mediated by forking. Adaptation, or development that *added* to infrastructure capabilities through modular development of the underlying source code, was also an important and influential mediator of self-organising.

These development forks enabled new capabilities by building upon the existing source code, which led to adaptation in the community. Adaptation thus enabled additional organising objectives through the creation of new infrastructure capabilities that built upon the existing source code without introducing incompatible elements.

As shown in Figure 4, the events of June 2011 attracted not only those with criminal or malicious intent, but also entrepreneurs, whom saw the potential for Bitcoin to be extended to delivering a range of financial and other services, most notably building credible merchant and retail solutions (F_5) and building wallets for use by Bitcoin users and investors (F_7).

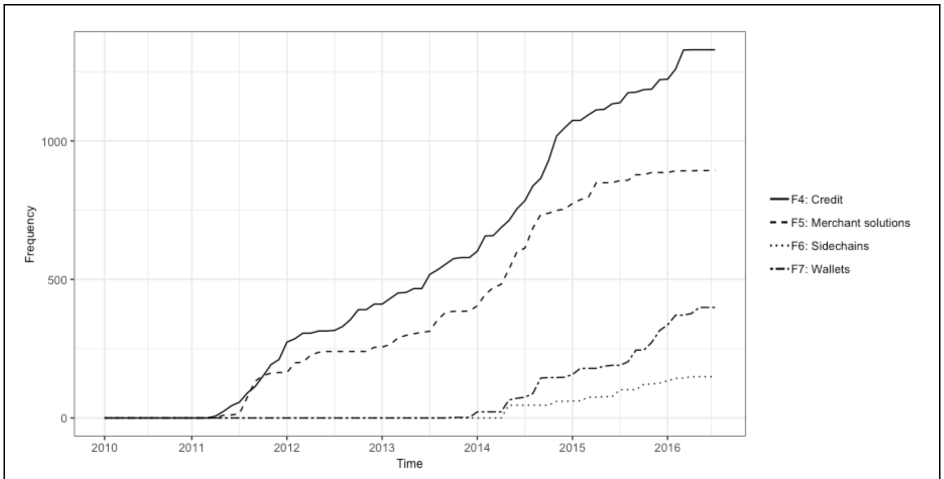

Figure 4. Activity Relating to Development Forks

Both giving credit (F_4) and developing sidechains (F_6) were dramatic changes to the blockchain widely discussed in the community in early 2014. However, while giving credit was incompatible with the infrastructure's capabilities and therefore could not be realised (without a hard fork), sidechains required mere addition to the source code—and therefore the organising objectives that they represented, namely the transfer of assets other than BTC through the blockchain, could be realised.

As these developments built upon the existing source code, they enabled the pursuit of new objectives, i.e. enabling retail and credit services, through the creation of start-ups that built complementary additions to the infrastructure. Among the ventures that started at this time are a number of Bitcoin exchanges[6] that are still successful in 2019, namely Bitstamp and Kraken, and the successful wallet Blockchain.info.[7] However, new adaptations were only accepted by the community under certain conditions, as one member of the community expressed:

Please understand that I have great respect for the work you've done. Your service is very well constructed and well-loved for good reason. While some of the things I've brought up might be improved with some tweaks here and there, much of it is simply the structural consequences of centralized services, trusted parties, web clients, etc…I don't think our community should take any actions which promote centralization or consolidation due to systemic risk if nothing else… I don't believe it should promote your wallet service either. - December 03, 2012, 03:02:53 PM

But entrepreneurship did not just extend to services that catered to the existing community. Instead, some of the adaptations mediated by development forks allowed the infrastructure to interface with other infrastructures, enabling inter-organisational linkages. One example is that of a Point-of-Sale (PoS) adaptation (F_5), which built both upon the underlying infrastructure and on other adaptations to extend the usefulness of the infrastructure and include new users:

We have released an update of our Pos [Point-of-Sale] software witch includes a module to connect and process payments trough Bitcoin-Qt wallet. So with this update, more than 3000 local busineses over the world who now are using Sysme Pos as point of sale software can have the tool to accept Bitcoins We hope that this will encourage them to accept Bitcoin as payment so this this project can make a step further. - June 05, 2013, 09:53:11 PM

Such an addition was not only useful on the level of infrastructure capabilities, it also served to include more stakeholders in supporting and engaging with Bitcoin's organising objectives. Not only did the creation of entirely new adaptations on the level of code, through development forks, mediate the pursuit of new organising objectives, they also expanded the number of stakeholders with an interest in the blockchain. Indeed, development forks like the creation of wallets and merchant services created new interdependencies by 1) integrating Bitcoin better into the established financial system by creating services for merchants and points at which bitcoins could be exchanged for fiat currencies like USD; and 2) linking Bitcoin and the blockchain to other related technologies, for instance in the creation of offline bitcoin wallets reliant on new pieces of hardware (e.g. Hurlburt and Bojanova 2014).

Among the adaptations that built upon the infrastructure are those that would allow, in an indirect way, the infrastructure to perform additional functions. While the blockchain's original architecture was intended as a proof-of-concept for the transfer of a currency, one adaptation allowed for the transfer and maintenance of a centralised database of assets other than that currency. In an adaptation known as a sidechain (F_6), developers built upon the blockchain to allow for an asset-agnostic transfer which interfaced with the blockchain:

The paper proposes two-way pegged sidechains as an extension mechanism for Bitcoin. The idea is that coins would be able to move from the Blockchain, to a sidechain, and then back again in a trustless way. This would allow sidechains to implement properties that are not feasible to implement on Bitcoin itself, while preserving the total number of Bitcoins. - September 10, 2015, 03:44:19 PM

---

[6] Analogous to currency exchanges, Bitcoin exchanges allow users to exchange Bitcoins for fiat currencies like the US Dollar or Euro, and vice versa

[7] Also analogous to a physical wallet, Bitcoin wallets served as a storage facility for Bitcoin access keys, whether online or offline

Though it represented a drastic development, this fork was still consistent with the underlying source code. This meant that, like in forks F_5 and F_7, the new objectives facilitated by the adaptation also relied on elements of the infrastructure's coded-in organising practices in order to function. Thus, adaptation was enabled (and constrained) by source code, through development forks. These development forks added to existing organising by attracting new users to the community, and by making the infrastructure itself able to support more things—thus changing its characteristics and capabilities on the code level.

Creative Adaptations

However, although some adaptations could be considered plain sailing, others were incompatible with the underlying infrastructure. Therefore, the underlying infrastructure could not support their pursuit. One example of this was in the creation of credit (F_4):

I'm not being misleading, the services offered by your current credit union could be extended to Bitcoins. This is a real institution with full financial services we're talking about so everything is on the table for the steering committee to discuss. Mortgages, Credit Cards, LOC, metals, multi currency handling (fiat and BTC), insurance etc. September 11, 2012, 04:28:10 PM

The fact that the creation of new BTC was limited by the blockchain's consensus mechanism meant that more could not be created just to support the leverage implicit in granting credit (Leland and Toft 1996). However, a creative credit-like adaptation was instead developed, namely the granting of peer-to-peer loans:

I am familiar with BTCPop, BTCJam, and Bitlendingclub (BLC). I love the concept of peer to peer lending as a way of disrupting traditional banking, and I love bitcoin. The two ideas work beautifully together - although unfortunately I don't think any company out there is doing it right yet. For example, loans are supposed to be tied to an interest rate, so that the sooner you repay the less interest you pay. This occurs with all traditional commercial debt, from mortgages to car loans to credit cards. All the bitcoin platforms, on the other hand, only calculate the debt over the term of the loan as a lump sum, so if you pay late or early it doesn't matter - you pay the same interest. So, for just one example, it makes no sense to refinance any debt to a lower interest rate. I'm hoping they do away with that soon or a competitor pops up which does. Peer to peer lending thrives on markets and it needs those options to be there. May 09, 2015, 04:31:21 PM

In early 2013, the development of peer-to-peer loans was seen as both a merit of the blockchain and a possible stumbling block to greater integration of the mainstream and Bitcoin financial systems. Altering the system so that it could issue credit would have required a substantial change to the blockchain infrastructure. Unlike the Bitcoin XT change (F_{10}) discussed below, the environmental pressure exerted in favour of giving credit was not (at the time) sufficient to drive the hard fork it would take to make it a reality. As such, it was a hypothetical hard fork that never reached the adoption threshold required to support a separate infrastructure with new capabilities. However, the environmental conditions that supported the idea of credit instead gave rise to another creative adaptation, namely the creation of unleveraged peer-to-peer loans.

M_3: Hard Forks as Bifurcation and Speciation

The most dramatic pattern of self-organising we observed was driven by a hard fork to the underlying code; that is, a fork in the code that created a new infrastructure that was both forwards- and backwards- incompatible with the existing infrastructure. If the community had adopted this 'forked' version of the code *en masse*, it would have become the dominant infrastructure. In fact, one might even say that it would become *the* infrastructure, while the versions of the code were either discarded or held by a less influential minority as a minority infrastructure. We refer to this pattern as 'bifurcation', as it resulted in two (or several) incompatible and competing infrastructures. The fact that a hard fork was necessary to pursue the creation of a new set of capabilities implied that, despite the substantial code-level similarities that they shared, at the level of code they are fundamentally incompatible.

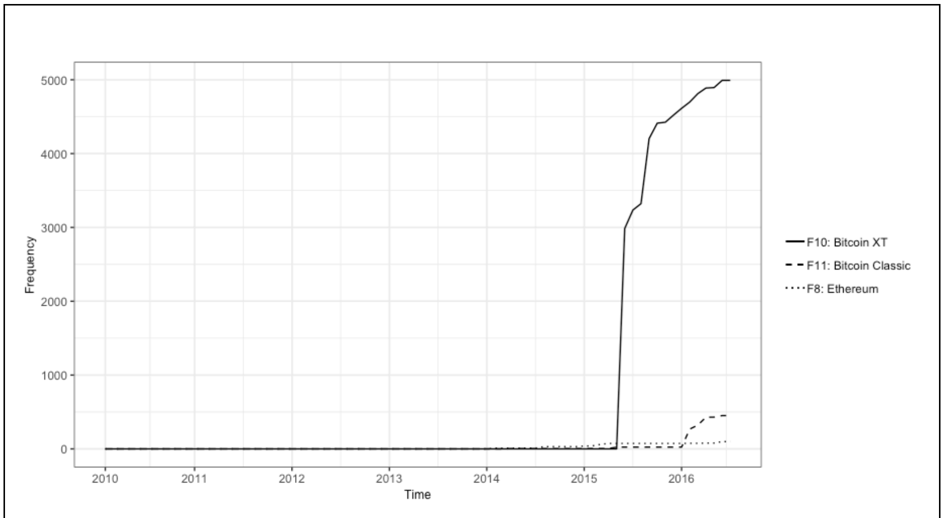

Figure 5. Activity Relating to Hard Forks

A particularly illustrative example is that of the Bitcoin XT (‘XT’, F_{10}): it not only resulted in a new infrastructure, but also the defensive pseudo-fork Bitcoin Core (F_9) described earlier. In June 2015 two prominent developers of the Bitcoin source code suggested that the sustainability of the blockchain infrastructure was in jeopardy. This was because, in late 2015, the infrastructure began to struggle to handle the number of Bitcoin transactions being conducted; there were delays and a backlog of transactions grew. Moreover, as demand was consistently increasing, the infrastructure needed better handling capacity—and the infrastructure did not have a built-in way to scale. In other words, the infrastructure began to fail (Hearn 2015). An infrastructural shift, known as Bitcoin XT, was proposed as a solution. This proposed solution would entail increasing the size of each block in the blockchain from 1mb to 4mb, which would require a fundamental change in the underlying source code:

As Bitcoin has grown, so have the blocks. Reasonable traffic projections indicate that as Bitcoin spreads via word of mouth, we will reach the limit of the current system [with a 1mb block size] sometime next year, or by 2017 at the absolute latest. And another bubble or press cycle could push us over the limit before even that. The result might not be pretty. So it is now time to raise the [block size] limit, or remove it entirely. - Mike Hearn, Aug 15, 2015

Bitcoin Classic (F_{11}), like XT before it, also proposed to increase the size of each block in the blockchain. Both aimed to facilitate processing a larger number of transactions more quickly. However, the blockchain protocol would also need to change fundamentally to allow for this. Moreover, increasing the block size would reduce the number of miners able to run the software (owing to issues around processing power), even as it increased the blockchain’s transaction handling capacity. The rules for how a change to the operating protocol of the blockchain could occur were enshrined in the source code. In essence, minor changes that created compatible versions of the software were easily dealt with, while major changes, like this hard fork, required active adoption beyond a certain threshold. Participants in the network, whether miners or entrepreneurs running platforms on the blockchain infrastructure, had to choose which version to run. Ultimately, the version that garnered the most support became the ‘real’ blockchain.

Opponents of this change labelled the original version of the infrastructure Bitcoin Core, and they argued that, among other things, XT was untried and may not scale well (with which XT proponents disagreed). They also argued that the change would make the project more centralised, putting more power in the hands of fewer miners—who could make more drastic changes in the future. OS communities often identify strongly with the underlying project (Ingram Bogusz and Morisse 2018; Ren et al. 2012), so controversial attempts to change it were often taken personally:

Bitcoin’s own former lead dev, Gavin, and his henchman Hearn are in the process of sabotaging Bitcoin from the inside. You will hear about it when their XT Trojan horse deploys its payload, and attempts to force us all to join their altcoin at Bitcoin’s expense. - July 9, 2015, 03:42:23 PM

This back-and-forth shows how the political environment affected how code changes were perceived and framed, which affected the community’s willingness to adopt new changes. In essence, they agreed with the need to do something, but argued that the technical shift to XT would change the social practices whereby the infrastructure operated by centralising control of the source code.

The blockchain source code ultimately bifurcated into in two forward- and backward-incompatible versions of the blockchain, within the same community. As miners did not adopt the XT version of the code in sufficient numbers, it never became the dominant version of the infrastructure. Moreover, swathes of the Bitcoin community also boycotted entrepreneurs and users who switched over to XT. Unlike with a development fork, wherein the new version of the infrastructure was compatible with those running the old version of the software, a hard fork meant that those running pre-fork versions of the infrastructure were unable recognise the changes.

A further hard fork, labelled Ethereum (F_8), revolved around alternative uses of blockchain-like infrastructures other than for Bitcoin, driven at least partly by the limitations the blockchain was seen to have at the time:

They can make an altcoin and voila, they'll have a blockchain. But there's no stronger blockchain than Bitcoin's. And if that's advertised enough, organizations will come for the blockchain and will have to use bitcoin for that. For the moment, the bitcoin blockchain is the most 'used' and it is the most secure but like I said "for the moment". However it is normal that in the future someone will build something better than the actual bitcoin (this is obvious... technology). - July 15, 2015, 07:56:16 PM

Users pointed both to the fact that Bitcoin transactions were slow (as we saw in the motivation for Bitcoin XT), and to the fact that the blockchain in its then-incarnation did not easily allow for the storage of other kinds of information, assets, or the execution of smart contracts. Although the originating objective behind Ethereum was to facilitate the use of blockchain technology for other kinds of applications, for instance smart contracts, the extent of the code-level changes involved and the threshold for adoption was sufficiently high that this generated a whole new species of blockchain with its own distinct community. The resulting Ethereum blockchain infrastructure, with its separate community and incompatible code base, was both operationally and ideologically distinct from the parent community, or a different species.

Having discussed how the three different types of forks outlined in the literature lead to four distinct patterns of self-organising, we turn to discussing how these findings inform emerging theory on blockchain infrastructures, specifically by relating them to generativity and self-organising.

Discussion and Contributions

This paper makes the following main contributions to our understanding of forking as a pattern of self-organising in blockchain infrastructures. First, it provides a vocabulary for forking as self-organising in blockchain infrastructures that extends the emerging body of research on the role of blockchain in organising (Avital et al. 2016; Beck et al. 2017, 2018; Catalini and Gans 2016). More specifically, it complements the literature on infrastructure evolution (Andersen and Ingram Bogusz 2017; Grisot et al. 2014; Hanseth and Lyytinen 2010; Henfridsson and Bygstad 2013; Tilson et al. 2010) by showing how forking in blockchain infrastructures leads to self-organising. Second, it explains how the environmental conditions surrounding forking attempts—specifically community adoption, heterogeneity of organising objectives, and code-level changes—enable the forking that mediates self-organising in blockchain infrastructures. Lastly, we discuss the implications of self-organising for blockchain design, forking itself, and technology-organisation alignment, known as ‘mirroring’.

Our findings describe the emergent consequences of the interplay between blockchain infrastructures and heterogeneous organising objectives in the case the Bitcoin blockchain. In particular, we zoom in on forking, thus far seen as a governance phenomenon within the blockchain community (Beck et al. 2018) and OS communities more generally (Nyman and Lindman 2013), as an important pattern that enables self-organising facilitated by the medium of computer code. Understanding the role of forking is not just important as organisations and societies become more digitalised, it is also important in light of how blockchain infrastructures enable increasingly autonomous organising, for instance through Distributed Autonomous Organisation (DAOs, Beck et al. 2016, 2018). This manner of organising is not only embedded in a distributed infrastructure, but challenges what we already know about how distributed and autonomous organising occurs across technological, geographical, industry, and organisational boundaries.

While the ramifications of distributed self-organising through blockchain infrastructures are potentially radical and far-reaching (Wörner et al. 2016), this paper focuses on the more narrow question of how self-organising has occurred in the past, specifically through a longitudinal study of the Bitcoin blockchain in its formative years, from October 2010 to June 2016.

We now turn to elaborating on, and discussing the implications of, these findings before concluding and discussing directions for future research.

Patterns of Forking as Self-Organising

In the following, we discuss how the three forking mechanisms discussed above, namely pseudo, development, and hard forks, led to different patterns of self-organising by enabling or constraining the pursuit organising objectives. Inspired by research framing Bitcoin as an ecosystem (Andersen and Ingram Bogusz 2017; Gandal et al. 2018; Wörner et al. 2016) and consistent with extant literature on digital infrastructure evolution (Hanseth and Lyytinen 2010; Henfridsson and Bygstad 2013), we have labelled these patterns of self-organising ‘variation’, ‘adaptation’, 'bifurcation', and ‘speciation’ as outlined in Table 5 and Figure 6. We now discuss how these patterns of self-organising emerge through forking, as well as how they link to the pursuit of organising objectives and related adoption practices.

Table 5. Patterns of Self-Organising

Pattern of Self-Organising	Description	Forks in Infrastructure Capabilities (F_n)	Shifts in Organising Objectives (O_n)
Variation *(M_1: Pseudo-forking)*	Incremental development of new organising objectives based on existing infrastructure capabilities	F_1: Darknet	O_1: From providing decentralised alternative to the financial system to illicit purposes for own gain
		F_2: Hacking	
		F_3: Scams	
		F_9: Bitcoin Core	O_7: Response to the Bitcoin XT hard fork (F_9)
Adaptation *(M_2: Development forking)*	Incremental development of new infrastructure capabilities and organising objectives	F_4: Credit	O_2: Adding capabilities of supplying financial services
		F_5: Merchant apps	O_3: Adding solutions for using Bitcoin as a payment infrastructure
		F_6: Sidechains	O_4: Temporarily mitigating the openness and decentralisation of Bitcoin while
		F_7: Wallets	O_5: Making it safe and convenient for individuals to store BTC
Bifurcation *(M_3: Hard forking)*	Radical divergence in both infrastructure capabilities and organising objectives taking place within the community	F_{10}: Bitcoin XT	O_6: Refactoring the Bitcoin infrastructure so that it can handle larger transaction volumes
		F_{11}: Bitcoin Classic	
Speciation *(M_3: Hard forking)*	Spin-off into separate technological infrastructure self-organised through a separate community	F_8: Ethereum	O_7: Building a new generation of blockchain infrastructure

Variation refers to the process whereby existing infrastructure capabilities are employed in pursuit of new organising objectives. Variation in organising objectives is thus facilitated by repurposing and reinterpreting existing technological capabilities through pseudo-forking. These variations connect unrelated organising objectives to the blockchain infrastructure as in the case of hacking (F_2), scams (F_3), and the re-labelling of the Bitcoin blockchain as Bitcoin Core (F_9). Variations on the purpose of existing capabilities do not require changes to the source code. Instead, the source code enables unexpected social practices in the form of new organising objectives.

Adaptation refers to the process by which supplementary infrastructure capabilities are generated on top of the infrastructure's existing capabilities, notably by adding code modules or creating supplementary, but compatible, capabilities. This process results in the infrastructure being used for new things, in new domains, and in an expansion of the blockchain user base. This results in compatible supplementary organising objectives operating within the existing infrastructure. Examples of this include the introduction of new entrepreneurial products like merchant (F_5) or wallet (F_7) applications. The added code allows not only for the pursuit of supplementary objectives, but also the creation of a supplementary organising and interactions reliant on the underlying blockchain infrastructure.

Bifurcation refers the process by which diverging organising objectives are generated in light of radical breaks with existing organising objectives, inconsistent with existing infrastructure capabilities. The organising objectives that are pursued in this way require fundamental (and incompatible) changes to the capabilities of the underlying blockchain infrastructure. For bifurcation to occur, code changes must supplant existing infrastructure capabilities, generating a new infrastructure that is technically incompatible with the pre-existing source code. Moreover, the social conditions under which such a hard fork occurs leads to the creation of sub-communities within the larger community, with new organising objectives, and not to the creation of a whole new community. Two examples of this, one realised and one unrealised, are the hard forks around BitcoinXT (F_{10}) and Credit-enabling leverage through the infrastructure (F_4). Both divided the existing community because they introduced centralisation to the blockchain infrastructure, a fundamental change to the underlying source code: Bitcoin XT did this by decreasing the number of miners who could run the blockchain protocol, while leverage would have required a third party who could control money supply. The resulting bifurcated blockchain infrastructures, thus, not only change the capabilities of the infrastructure, but also who could control the infrastructure: a departure from the distributed, non-hierarchical organisation of the parent ('Core'. F_9) infrastructure.

Speciation refers to cases of bifurcation where changes in infrastructure capabilities and organising objectives are so divisive that they spin off into completely separate blockchain infrastructures with their own sets of capabilities, objectives and—crucially—communities. An example of speciation in the Bitcoin infrastructure can be seen in the emergence of Ethereum (F_8), which, despite being a fork to the Bitcoin blockchain, operates independently.

The impact of code forking, both as reflected in additions to the blockchain infrastructure and in how they are discussed, are relatively clear and distinct. Both variation and adaptation, generated by pseudo- and development forks respectively, result from compatible variations on, or additions to, existing code. In contrast, both bifurcation and speciation, generated by hard forks, result from incompatible additions to the existing source code. The overall patterns of forking and adoption are summarised in Figure 6 and reflected analytically in Table 6.

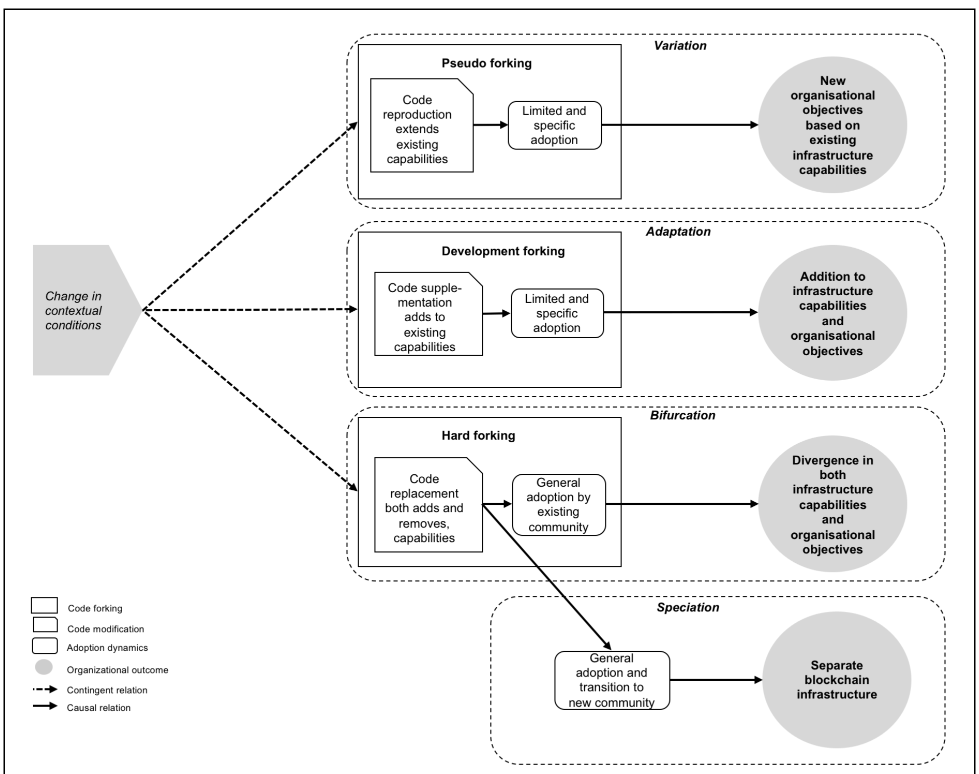

Figure 6. Overview of the Patterns of Self-Organising

Having established the consequences of forking in blockchain infrastructures, the question arises of *how* the different forks mediate these different self-organising patterns. In brief, distributed blockchain infrastructures' self-organising is generated by 1) heterogeneity of organising objectives and 2) specific patterns of community adoption. These are discussed further below.

Heterogeneous Objectives and Patterns of Adoption

Research elsewhere, both within OS communities more generally (Choi et al. 2015) and the Bitcoin community specifically (Ingram Bogusz and Morisse 2018) has shown that online communities are seldom comprised of groups of individuals with homogenous shared objectives and ideologies. Instead, these communities have multiple, heterogeneous, and potentially conflicting objectives. Indeed, as our analysis above shows, the larger the community, the more potential variation—and potential for conflict—results. While recent research on blockchain generativity has shown how "…disagreements can be resolved in a decentralised manner if users initiate forks by copying existing code and developing it further according to their goals" (Beck et al. 2018: 27), the process by which this occurs, and the role of the code as infrastructure, has yet to be studied.

Our analysis of the Bitcoin blockchain shows that in the face of heterogeneity, it is the pre-existing content of the code—what we have called infrastructure capabilities—that determines whether originating objectives give rise to the pursuit of compatible and complementary objectives (which avoid further dispute), or to diverging ones. Where existing code is either flexible or does not prevent a proposed objective, what results are (a set of) emerging objectives that are limited in scope, compatible with the existing infrastructure, and may require specific adoption in order to have any effect at all (see Table 6). This is true whether the objectives pursued are accepted as legitimate by the community, or not. Indeed, the resulting adaptation and variation cannot be severed from the originating infrastructure—and the code-level additions (adaptation), and new social employment of existing code (variation) cannot be prevented or sanctioned by the community. Instead, opponents can avoid adoption—a limited form of sanction (Feller et al. 2008).

Table 6. Self-Organising in Blockchain Infrastructures

	Homogenous organising objectives	*Heterogeneous organising objectives*
General adoption of infrastructure capabilities *(Adoption across entire community)*	**Bifurcation** Divergence within the organisation due to radical self-organising (Hard fork)	**Speciation** Spin-off into new organising structure (e.g. Ethereum) (Hard fork)
Specific adoption of infrastructure capabilities *(Adoption by a specific sub-group within or outside the community)*	**Adaptation** Organising around specific objectives, business cases etc. (Development fork)	**Variation** Organising to take advantage of or repurpose blockchain infrastructure for other ends (Pseudo-fork)

However, the capabilities of the code may also generate divergent and heterogeneous paths within the same infrastructure (e.g. Bitcoin XT), or the creation of distinct and incompatible blockchain infrastructures (e.g. Ethereum). The key differentiator for the realisation of organising objectives is the scope of adoption, where adoption of a particular fork in specific sub-groups within the community or general adoption across the entire community influences the actual pattern of self-organising. Table 6 illustrates the role of adoption in generating different patterns of self-organising, and its link to homogenous or heterogeneous organising objectives.

Specific Adoption

The patterns of adoption that generate these changes, specifically the extent of code adoption, also affect how self-organising progresses, and what its outcome(s) are. In the case of variation and adaptation, adoption happens pursuant to different patterns of organising objective shifts, namely heterogeneous objectives given existing code (variation), and the creation of supplementary code by those with relatively homogenous organising objectives (adaptation). Moreover, specific adoption of a variation or adaptation is required, not general adoption of a new (blockchain) infrastructure. That is, as there are no internal limitations in the infrastructure that prevent the realisation of new capabilities and associated objectives. What transforms code-level capabilities into the realisation of new objectives is the adoption of the specific reproduction or repurposing of code that has occurred, not adoption of a larger code base.

General Adoption

In contrast, the patterns of code replacement, diverging organising objectives, and ultimate adoption in the face of hard forks differ substantially. Although all hard forks are prompted by environmental changes that motivate new objectives that are incompatible with the existing infrastructure, patterns around organising objectives and adoption influence the degree of self-organising and divergence that results.

Where there is commitment to the existing capabilities of the infrastructure within the community, but a desire to create additional incompatible capabilities, for instance faster processing power (as in Bitcoin XT), we see agreement (homogeneity) between those who would fork and those who remain behind. As a result, the forwards- and backwards-incompatible code changes that are needed for the fork occur within the fold of the existing community. For the emerging new fork to become the primary infrastructure, the bifurcation generated, in turn, requires a substantial *general* adoption within that community. In the absence of this general adoption, a hard fork could become an alternative infrastructure (with adoption by a minority), or a failed hard fork (without significant adoption).

In contrast, where the existing objectives of the existing community and a new set of objectives share little common ground, for instance where the new objectives revolve around smart contracts more generally, not just currency transfers (as in Ethereum), what results is the creation of new capabilities through both a separate species of infrastructure and a separate community. As with bifurcation, the resulting new infrastructure could eclipse its parent with sufficiently large adoption. However, given the heterogeneous objectives involved, what would be generated is either a whole new species of infrastructure with substantial adoption, or a failed hard fork.

In essence, existing code supports and reinforces the *status quo*, and changes to it require significant resources and mobilisation on the part of would-be forkers. Consequently, self-organising through hard forks could pan out in one of four ways: 1) the hard fork could be absorbed by the existing infrastructure, which would require general adoption by a majority of users; 2) the hard fork could generate a whole new community; 3) the hard fork could generate a minority infrastructure; and 4) the hard fork could fail due to lack of general adoption. These four possible outcomes stem from individual and collective responses to an otherwise technical change: both the nature of the organising objectives at play, and the threshold of adoption are relevant to what the generative consequences of a code forking are.

Implications

Overall, this paper conceptualises how forking in blockchain infrastructures generates self-organising. More specifically, it outlines the means by which forking facilitates the pursuit of organising objectives, and the role of: 1) the underlying infrastructure; 2) the scale of the code changes; 3) individual objectives; and 4) collective adoption, whether specific or general, in this pursuit. By identifying four patterns of self-organising and their antecedents, this paper outlines a framework for assessing the potential consequences of specific forking events. Moreover, its conceptualisation of the antecedents of these patterns; homogenous and heterogeneous objectives, and thresholds of adoption needed for certain outcomes, give guides to would-be blockchain infrastructure designers around what to consider when developing a governance model. Unpacking these different mechanisms gives us insight into how a blockchain infrastructure could be governed to facilitate—or prevent—the different outcomes described in Table 6.

Generativity and Blockchain Design

When it comes to blockchain design, blockchain infrastructures, unlike platforms and applications, do not have strict boundaries and cannot be defined through a specific set of functions or modules (Tilson et al. 2010). Instead, their boundaries are defined through use practices, evident in the pursuit of organising objectives and adoption in the generation of new patterns of self-organising—and the maintenance of old ones. Our findings reinforce the idea that the existence—and design—of an infrastructure relies on continued use, maintenance and self-organising by a distributed group of users (Henfridsson and Bygstad 2013). Indeed, we build on these ideas to show how use and maintenance support the emergence of different patterns of self-organising.

In particular, lowering the transaction costs of collaboration (Catalini and Gans 2016) facilitates the emergence of self-organising through a distributed process in which multiple users contributes to an underlying infrastructure (Yoo et al. 2012). This builds on findings related to committal structures in OS communities, where it was found that decentralisation allowed for better performance in situations with high levels of uncertainty (AlMarzouq et al. 2015). Here, there were high levels of uncertainty particularly in the case of hard forks: changes were often hotly contested at the level of adoption and use. Indeed, the Bitcoin community, like other OS communities, discouraged rogue changes through political ideology and community backlash (Dahlander and Magnusson 2005; Ingram Bogusz and Morisse 2018; Kirsch 1997). The open nature of the code, however, meant that on a technical level forking could not be prevented—and thus the patterns of self-organising that result could not be prevented (but may be limited or mitigated). However, consensus mechanism affected how open a blockchain was to generativity. It did this by determining the (quantitative or qualitative) threshold for when a fork generated variation and adaptation, or bifurcation and speciation. Fine-tuning a consensus mechanism through design therefore provides an avenue for the enabling or constraining of generativity—for better or worse (Yoo et al. 2012).

These findings thus have implications when it comes to designing for generativity in blockchain infrastructures. Extant research has suggested that the design of a digital infrastructure requires accounting for dynamic complexity (Hanseth and Lyytinen 2010), but the very nature of such complexity makes anticipating possible generativity tricky. However, conceptualisation of the specific routes that generativity could take gives designers concrete areas for consideration when designing blockchain infrastructures. Some specific questions include the following. First, when it comes to what we have called speciation and bifurcation, what are the code-level thresholds involved that will lead to changes to the fabric of the infrastructure through a hard fork, and do such changes require general adoption of a whole infrastructure? Second, when it comes to complementary areas of generativity, i.e. adaptation and variation, are there limits to where and how additions or variations can be added to existing objectives, and what are the specific patterns of adoption around the code for the new objectives. These considerations, moreover, evolve as change is generated; thus the changes to source code and code use outlined in Figure 6 are relational in nature and the maintenance of the community infrastructure necessitates constant interaction.

Forking

Although the aim of this paper is not to contribute to the forking literature, but to analyse the role of forking in the blockchain, we have nevertheless built on literatures on forking in order to better understand a new kind of digital infrastructure, the blockchain.

In particular, while extant research has referred to forking as a governance phenomenon in its own right (e.g. Meeker 2008; Nyman 2015), we show how forking can be part of a larger process of self-organising, given certain antecedents and in certain contexts. Moreover, a single kind of hard fork in the presence of different objectives and code capabilities can give rise to completely different self-organising processes. In this case, a hard fork can lead to either bifurcation, where a new sub-community with new objects emerges within the existing community, or to speciation, which is the creation of a whole new blockchain infrastructure and community that does not compete directly with the parent infrastructure because it has distinct technological capabilities and organising objectives.

These contributions show not only that code itself is pivotal in determining, at least in the case of blockchain infrastructures, what organising processes result, but also that there is a complex dynamic between code capabilities and organising objectives that could lead to outcomes previously not seen or theorised within the forking literatures.

Blockchain and Generative Mirroring

Ultimately, extant research sees changes in organising as generating infrastructural evolution, rather than vice versa (e.g. Ciborra 2000; Eaton et al. 2015). By showing how self-organising is mediated by code forking in the blockchain, we contribute a more nuanced understanding of the role of the digital in its own self-organising: while intended objectives and adoption of the code are a vital part of organising in this context, possibilities are also defined according to the code-level composition of an infrastructure, as seen in this specific case of the blockchain infrastructure.

This paper further begins to build a conceptualisation of self-organisation in digital infrastructures, including what we call ‘generative mirroring’. Extant studies of mirroring have shown how organisations can mirror the characteristics of technological artefacts upon which they rely, in order to generate performance gains. However, this *ex ante* design does not reflect either the dynamism between digital infrastructures and distributed users and developers, or how the relationship between the technology and organising emerges through interaction (both social and at the level of code) and adoption. Instead, the distributed nature of control in blockchain infrastructures, i.e. through consensus mechanisms, leads not only to generativity, but also to generative mirroring, as the infrastructure and its community evolve in tandem. In this paper, we identify forking as one such mechanism of generative mirroring, but there may well be others—especially as automation proceeds apace. This finding adds to the emerging body of literature in generativity in digital infrastructures (Hanseth and Lyytinen 2010; Henfridsson and Bygstad 2013; Koutsikouri et al. 2018; Tilson et al. 2010; Um et al. 2015).

The findings presented in this paper have several implications for understanding organising in the blockchain (Beck et al. 2018; Böhme et al. 2015; Iansiti and Lakhani 2017), especially public and permissionless blockchain infrastructures, like Bitcoin, without *a priori* governing structures. Existing research has pointed out that organising around blockchain infrastructures is still in need of further research (Lindman et al. 2017), and our findings contribute in this area. A summary of the implications discussed above is presented in Table 7.

Table 7. Implications of Self-Organising in Blockchain Infrastructures

	Contribution	**Implications**
Blockchain Design	Both use practices and code capabilities delimit otherwise porous infrastructure boundaries	The existence—and evolution—of an infrastructure relies on the continued practice of use, maintenance and self-organisation by a distributed group of users
	Lowered transaction costs of collaboration facilitates the emergence of self-organisation through forking	Tweaking the thresholds for collaboration costs, for instance through the content of a consensus mechanism, likely to impact how generative a blockchain infrastructure becomes
Forking	Forking not just a governance phenomenon, but a dynamic between code capabilities and objectives	The same kind of hard fork, in the presence of different objectives and code capabilities, can give rise to completely different forms of self-organising
Blockchain Generativity	While intended objectives and adoption of the code are a vital part of organising, the possibilities open are also defined according to the code-level capabilities of a blockchain infrastructure	Challenges the existing view of digital infrastructures, which points to the centrality of human activities to organising, rather than the interaction of these activities with the capabilities of the infrastructure itself
	Generative Mirroring	A blockchain infrastructure and its community evolve in tandem, through interactions between its technology and community with organising emerging through interaction (both social and at the level of code) and adoption

Conclusion and Directions for Future Research

Digital infrastructures are gaining significance within both existing and emerging organisations (von Briel et al. 2018; Nambisan et al. 2018; Tilson et al. 2010) and, with the rise of blockchain technology, autonomous organising is less the stuff of science fiction, and increasingly a reality. Our examination of self-organising in the Bitcoin infrastructure contributes to conceptualising how blockchain infrastructures self-organise as a radical technological innovation that shapes—and is shaped by—organising objectives. Understanding the patterns underlying self-organising informs the growing body of knowledge on blockchain as an organising phenomenon, not just a technological one (Beck et al. 2018; Catalini and Gans 2016; Iansiti and Lakhani 2017). This paper provides an examination of how self-organising occurs in blockchain infrastructures, and the role of 1) the underlying infrastructure; 2) the scale of the code changes; 3) individual objectives; and 4) collective adoption, whether specific or general, in these processes. By identifying four patterns of self-organising and their antecedents, this paper outlines a framework for assessing the potential consequences of specific forking events. It builds an understanding of how the specific characteristics of blockchain technology lead to new patterns of self-organising, and the role of forking in those patterns.

By building an understanding of forking as a mode of self-organising in blockchain infrastructures, this paper contributes to extant research on blockchain infrastructures in the following ways: First, it identifies and conceptualises a new mode of organising mediated by blockchain infrastructures where forking is a necessary (but not sufficient) condition for the realisation of new organising objectives and practices. Second, it proposes, substantiates, and empirically identifies the concept of forking to show how, and when, code development practices combine into a pattern of self-organising. In so doing, it provides a vocabulary for describing the different patterns of self-organisation in blockchain infrastructures; variation, adaptation, bifurcation, and speciation.

Blockchain infrastructures are, in many ways, the harbingers of infrastructures to come: distributed control, fluid technological capabilities, and openness to manipulation by a large and heterogeneous community are increasingly commonplace (Beck et al. 2018; Hanseth and Lyytinen 2010). This is partly because of technological advances, and also because community members (whether users or miners) themselves hold the skills, and tools, to support—and destroy—a digital infrastructure through code forking and adoption (Kallinikos et al. 2013; Rullani and Haefliger 2013). Indeed, the fact that code-level changes and associated objectives are often hotly contested, both this paper and previous research in OS communities show that forking is sometimes unavoidable. Understanding its role in the design and organisation of a digital infrastructure like the blockchain therefore holds value as such infrastructures become more common—and indeed vital to modern societies. We therefore encourage future examination of the role of code, and code forking in particular, in organisational change and self-organising. We further encourage research into patterns of self-organising other than forking, whether in blockchain infrastructures or next generation distributed digital infrastructures. Lastly, we encourage empirical studies that improve our understanding of generative mirroring.

References

Aanestad, M., and Jensen, T. B. 2011. “Building nation-wide information infrastructures in healthcare through modular implementation strategies,” *Journal of Strategic Information Systems* (20:2), Elsevier B.V., pp. 161–176.

AlMarzouq, M., Grover, V. and Thatcher, J.B., 2015. "Taxing the development structure of open source communities: An information processing view." *Decision Support Systems*, *80*, pp.27-41.

Andersen, J. V., and Ingram Bogusz, C. 2017. “Patterns of Self-Organising in the Bitcoin Online Community : Code Forking as Organising in Digital Infrastructure,” in *Proceedings of the Thirty eighth International Conference on Information Systems (ICIS 2017)*, Seoul, pp. 1–20.

Anderson, P. 2012. “Complexity Theory and Organization Science,” *Organization Science* (10:3), pp. 216–232.

Austin, R. D., Devin, L., and Sullivan, E. E. 2011. “Accidental Innovation: Supporting Valuable Unpredictability in the Creative Process,” *Organization Science* (23:5), pp. 1505–1522.

Avital, M., King, J. L., Beck, R., Rossi, M., and Teigland, R. 2016. “Jumping on the Blockchain Bandwagon : Lessons of the Past and Outlook to the Future Panel,” *ICIS 2016 Proceedings*, pp. 1–6.

Bano, S., Sonnino, A., Al-Bassam, M., Azouvi, S., McCorry, P., Meiklejohn, S., and Danezis, G. 2017. “Consensus in the Age of Blockchains,” *arXiv preprint*.

Barrett, M., Oborn, E., and Orlikowski, W. J. 2016. “Creating value in online communities: the sociomaterial configuring of strategy, platform, and stakeholder engagement,” *Information Systems Research* (27:4), Informs, pp. 704–723.

Bechky, B. A. 2006. “Talking about machines, thick description, and knowledge work,” *Organization Studies* (27:12), Sage Publications, pp. 1757–1768.

Beck, R., Avital, M., Rossi, M., and Thatcher, J. B. 2017. “Blockchain Technology in Business and Information Systems Research,” *Business and Information Systems Engineering* (59:6), Springer Fachmedien Wiesbaden, pp. 381–384.

Beck, R., Czepluch, S., Lollike, N., and Malone, S. 2016. "Blockchain - The Gateway to Trust-Free Chryptographic Transactions," in *ECIS Proceedings*, pp. 1–14.

Beck, R., and Müller-Bloch, C. 2017. "Blockchain as Radical Innovation: A Framework for Engaging with Distributed Ledgers," in *Proceedings of the 50th Hawaii International Conference on System Sciences*, Manoa

Beck, R., Müller-Bloch, C., and King, J. L. 2018. "Governance in the Blockchain Economy: A Framework and Research Agenda," *Journal of the Association for Information Systems* (forthcoming).

Beckert, J. 1999. "Agency, Entrepreneurs, and Institutional Change. The Role of Strategic Choice and Institutionalized Practices in Organizations," *Organization Studies* (20:5), pp. 777–799.

Berente, N., Seidel, S., and Safadi, H. 2018. "Data-Driven Computationally-Intensive Theory Development " *Information Systems Research* (forthcoming).

Blei, D. M., Ng, A. Y., and Jordan, M. I. 2003. "Latent dirichlet allocation," *The Journal of Machine Learning Research* (3:1), pp. 993–1022.

Böhme, R., Christin, N., Edelman, B., and Moore, T. 2015. "Bitcoin: Economics, Technology, and Governance," *Journal of Economic Perspectives* (29:2), pp. 213–238.

Braa, J., Hanseth, O., Heywood, A., Woinshet, M., and Shaw, V. 2007. "Developing Health Information Systems in Developing Countries : the flexible Standards Strategy," *Management Information Systems Quarterly* (31:Suppl 1), pp. 381–402.

Brenig, C., Schwarz, J., and Rückeshäuser, N. 2016. "Value of Decentralized Consensus Systems - Evaluation Framework," *ECIS 2016 Proceedings*, pp. 1–18.

von Briel, F., Davidsson, P., and Recker, J. 2018. "Digital technologies as external enablers of new venture creation in the it hardware sector," *Entrepreneurship: Theory and Practice* (42:1), pp. 47–69.

Broadbent, M., and Weill, P. 1997. "Management by maxim: How business and IT managers can create IT infrastructures," *Sloan Management Review* (38:3), pp. 77–92.

Burton, N., and Galvin, P. 2018. "When do product architectures mirror organisational architectures? The combined role of product complexity and the rate of technological change," *Technology Analysis and Strategic Management* (30:9), Taylor & Francis, pp. 1057–1069.

Bygstad, B. 2010. "Generative mechanisms for innovation in information infrastructures," *Information and Organization* (20:3), Elsevier, pp. 156–168.

Callon, M. 1986. "Some elements of a sociology of translation: domestication of the scallops and the fishermen of St Brieuc Bay," in *Power, action and belief: A new sociology of knowledge?*J. Law (ed.) (Vol. 32), Routledge, pp. 196–233.

Campbell-Verduyn, M. 2018. *Bitcoin and Beyond* (M. Campbell-Verduyn, ed.), London: Routledge.

Catalini, C., and Gans, J. S. 2016. "Some Simple Economics of the Blockchain," *SSRN* Online.

Chapron, G. 2017. "The environment needs cryptogovernance," *Nature* (545), p. 403.

Child, J. 1997. "Strategic Choice in the Analysis of Action, Structure, Organizations and Environment: Retrospect and Prospect," *Organization Studies* (18:1), pp. 43–76.

Chiravuri, A., Nazareth, D., and Ramamurthy, K. 2011. "Cognitive Conflict and Consensus Generation in Virtual Teams During Knowledge Capture: Comparative Effectiveness of Techniques," *Journal of Management Information Systems* (28:1), pp. 311–350.

Choi, N., Chengalur-Smith, I., and Nevo, S. 2015. "Loyalty, Ideology, and Identification: An Empirical Study of the Attitudes and Behaviors of Passive Users of Open Source Software," *Journal of the Association for Information Systems* (16:8), pp. 674–706.

Chuang, J., Manning, C. D., and Heer, J. 2012. "Termite : Visualization Techniques for Assessing Textual Topic Models Categories and Subject Descriptors," *International Conference on Advanced Visual Interfaces (AVI)*.

Ciborra, C. 2000. *From control to drift: the dynamics of corporate information infastructures*, Oxford University Press on Demand.

Dabbish, L., Stuart, C., Tsay, J., and Herbsleb, J. 2012. "Social coding in GitHub: transparency and collaboration in an open software repository," in *Proceedings of the ACM 2012 conference on computer supported cooperative work*, ACM, pp. 1277–1286.

Dahlander, L., and Magnusson, M. G. 2005. "Relationships between open source software companies and communities: Observations from Nordic firms," *Research Policy* (34:4), pp. 481–493.

Drazin, R., and Sandelands, L. 1992. "Autogenesis - A Perspective on the Process of Organizing," *Organization Science* (3:2), pp. 230–249.

Eaton, B., Elaluf-Calderwood, S., Sørensen, C., and Yoo, Y. 2015. "Distributed Tuning of Boundary

Resources: the Case of Apple's Ios Service System," *MIS Quarterly* (39:1), pp. 217–243.
Edwards, P. N., Jackson, S. J., Bowker, G. C., and Knobel, C. 2007. "Understanding infrastructure: Dynamics, tensions, and design.," *Report of a Workshop on "History & Theory of Infrastructure: Lessons for New Scientific Cyberinfrastructures"*.
Fang, Y., and Neufeld, D. 2009. "Understanding Sustained Participation in Open Source Software Projects," *Journal of Management Information Systems* (25:4), pp. 9–50.
Feller, J., Finnegan, P., Fitzgerald, B., and Hayes, J. 2008. "From peer production to productization: A study of socially enabled business exchanges in open source service networks," *Information Systems Research* (19:4), pp. 475–493.
Gandal, N., Hamrick, J., Moore, T., and Oberman, T. 2018. "Price Manipulation in the Bitcoin Ecosystem," *Journal of Monetary Economics*, Elsevier B.V.
Gaskin, J., Berente, N., Lyttinen, K., and Yoo, Y. 2014. "Towards Generalizable Sociomaterial Inquiry: A Computational Approach for Zooming In and Out of Sociomaterial Routines," *MIS Quarterly* (38:3), pp. 849–871.
Gawer, A. and Cusumano, M.A., 2014. Industry platforms and ecosystem innovation. *Journal of Product Innovation Management*, *31*(3), pp.417-433.
Glaser, B. G., and Strauss, A. L. 1999. *The Discovery of Grounded Theory: Strategies for qualitative research*, New Brunswick: AldineTransaction.
Grisot, M., Hanseth, O., and Thorseng, A. A. 2014. "Innovation of, in, on infrastructures: Articulating the role of architecture in information infrastructure evolution," *Journal of the Association for Information Systems* (15:4), pp. 197–219.
Guerini, M., and Moneta, A. 2017. "A method for agent-based models validation," *Journal of Economic Dynamics and Control* (82), Elsevier B.V., pp. 125–141.
Hanseth, O., Jacucci, E., Grisot, M., and Aanestad, M. 2006. "Reflexive Standardization: Side Effects and Complexity in Standard Making," (30), pp. 563–581.
Hanseth, O., and Lyytinen, K. 2010. "Design theory for dynamic complexity in information infrastructures: the case of building internet," *Journal of Information Technology* (25:1), pp. 1–19.
Hanseth, O., and Monteiro, E. 1997. "Inscribing behaviour in information infrastructure standards," *Accounting, Management and Information Technologies* (7:4), pp. 183–211.
Hearn, M. 2015. "Why is Bitcoin forking? A tale of differing visions," (available at https://medium.com/faith-and-future/why-is-bitcoin-forking-d647312d22c1#.8dq5fi3we).
Hedman, J., Srinivisan, N., and Lindgren, R. 2013. "Digital Traces of Information Systems: Sociomateriality Made Researchable," in *Thirty Fourth International Conference on Information Systems, Milan 2013*, pp. 809–830.
Henfridsson, O., and Bygstad, B. 2013. "The generative mechanisms of digital infrastructure evolution," *MIS Quarterly* (37:3), pp. 907–931.
Holland, J. H. 1992. "Complex Adaptive Systems," *Deadalus* (121:1), pp. 17–30.
Holland, J. H. 1995. *Hidden order: How adaptation builds complexity*, BOOK, Basic Books.
Howison, J., and Crowston, K. 2014. "Collaboration through open superposition: a theory of the Open Source way," *MIS Quarterly* (38:1), pp. 1–11.
Howison, J., Wiggins, A., and Crowston, K. 2011. "Validity Issues in the Use of Social Network Analysis with Digital Trace Data," *Journal of the Association for Information Systems* (12:12), pp. 767–797.
Hurlburt, G. F., and Bojanova, I. 2014. "Bitcoin: Benefit or curse?," *IT Professional* (16:3), IEEE, pp. 10–15.
Iannacci, F. 2010. "When is an information infrastructure? Investigating the emergence of public sector information infrastructures," *European Journal of Information Systems* (19:1), pp. 35–48.
Iansiti, M., and Lakhani, K. 2017. "The truth about blockchain - It will take years to transform," *Harvard Business Review* (February).
Ingram Bogusz, C., and Morisse, M. 2018. "How infrastructures anchor open entrepreneurship: The case of Bitcoin and stigma," *Information Systems Journal* (28:6), pp. 1176–1212.
Jones, Q., Ravid, G., and Rafaeli, S. 2004. "Information overload and the message dynamics of online interaction spaces: A theoretical model and empirical exploration," Information Systems Research (15:2), pp. 194–210.
Kallinikos, J., Aaltonen, A., and Marton, A. 2013. "The Ambivalent Ontology of Digital Artifacts," *MIS Quarterly* (34:2), pp. 213–231.
Kartseva, V., Hulstijn, J., Gordijn, J., and Tan, Y.-H. 2010. "Control patterns in a health-care network," *European Journal of Information Systems* (19:3), pp. 320–343.

Kirsch, L. S. 1997. “Portfolios of control modes and IS project management,” *Information Systems Research* (8:3), pp. 215–239.
Koutsikouri, D., Lindgren, R., Henfridsson, O., and Rudmark, D. 2018. “Extending Digital Infrastructures: A Typology of Growth Tactics,” *Journal of the Association for Information Systems* (19:10), pp. 1001–1019.
Kuk, G., and Janssen, M. 2013. “Assembling infrastructures and business models for service design and innovation,” *Information Systems Journal* (23:5), pp. 445–469.
Latour, B. 1987. *Science in action: How to follow scientists and engineers through society*, Harvard university press.
Lave, J., and Wenger, E. 1991. *Situated learning: Legitimate peripheral participation*, Cambridge university press.
Lee, J., and Berente, N. 2012. “Digital Innovation and the Division of Innovative Labor: Digital Controls in the Automotive Industry,” *Organization Science* (23:5), pp. 1428–1447.
Leising, M. 2017. "The Ether Thief", Available online at https://www.bloomberg.com/features/2017-the-ether-thief/
Leland, H. E., and Toft, K. B. 1996. “Optimal capital structure, endogenous bankruptcy, and the term structure of credit spreads,” *The Journal of Finance* (51:3), pp. 987–1019.
Leonardi, P. M., and Barley, S. R. 2008. “Materiality and change: Challenges to building better theory about technology and organizing,” *Information and Organization* (18:3), pp. 159–176.
Lindberg, A., Berente, N., Gaskin, J., and Lyytinen, K. 2016. “Coordinating Interdependencies in Online Communities: A Study of an Open Source Software Project,” *Information Systems Research* (27:4), pp. 751–772.
Lindman, J., Rossi, M., and Tuunainen, V. K. 2017. “Opportunities and risks of Blockchain Technologies in payments– a research agenda,” *HICSS 2017 Proceedings*, pp. 1533–1542.
Markus, M. L., & Robey, D. 1983. The organizational validity of management information systems. *Human Relations*, *36*(3), 203-225.
Meeker, H. J. 2008. *The open source alternative: understanding risks and leveraging opportunities*, John Wiley & Sons.
Miles, M. B., and Huberman, A. M. 1994. *Qualitative data analysis: An expanded sourcebook*, Sage.
Mol, A., and Law, J. 2002. *Complexities: an introduction,* Duke University Press.
Nærland, K., Müller-bloch, C., Beck, R., and Palmund, S. 2017. “Blockchain to Rule the Waves - Nascent Design Principles for Reducing Risk and Uncertainty in Decentralized Environments Abstract,” *ICIS 2017 Proceedings*, pp. 1–16.
Nakamoto, S. 2008. “Bitcoin: A peer-to-peer electronic cash system,” (Vol. 1).
Nambisan, S., Siegel, D., and Kenney, M. 2018. “On open innovation, platforms, and entrepreneurship,” *Strategic Entrepreneurship Journal* (12:3), John Wiley & Sons, Ltd, pp. 354–368.
Nan, N. 2011. “Capturing Bottom-up Information Technology Use Processes: A Complex Adaptive Systems Model,” *MIS Quarterly* (35:2), pp. 505–532.
Nyman, L. 2015. “Understanding Code Forking in Open Source Software,” Hanken School of Economics.
Nyman, L., and Lindman, J. 2013. “Code Forking, Governance, and Sustainability in Open Source Software,” *Technology Information Management* (January), pp. 7–12.
O’Mahony, S., and Ferraro, F. 2007. “The emergence of governance in an open source community,” *Academy of Management Journal* (50:5), 1079–1106.
Peters, G. W., and Panayi, E. 2015. “Understanding Modern Banking Ledgers through Blockchain Technologies: Future of Transaction Processing and Smart Contracts on the Internet of Money,” *SSRN Online*.
Pettigrew, A. M., Woodman, R. W., and Cameron, K. S. . 2001. “Studying Organizational Change and Development: Challenges for Future Research,” *Academy of Management Journal* (44:4), pp. 697–713.
Pipek, V., and Wulf, V. 2009. “Infrastructuring: Toward an Integrated Perspective on the Design and Use of Information Technology,” *Journal of the Association for Information Systems* (10:May 2009), pp. 447–473.
Raymond, E. 1999. “The cathedral and the bazaar,” *Knowledge, Technology & Policy* (12:3), pp. 23–49.
Ren, Y., Harper, F. M., Drenner, S., Terveen, L., Kiesler, S., Riedl, J., and Kraut, R. E. 2012. “Building member attachment in online communities: Applying theories of group identity and interpersonal bonds,” *Mis Quarterly* (36:3), pp. 841–864.

Robles, G., and González-Barahona, J. M. 2012. “A comprehensive study of software forks: Dates, reasons and outcomes,” in *IFIP International Conference on Open Source Systems*, pp. 1–14.
Rullani, F., and Haefliger, S. 2013. “The periphery on stage: The intra-organizational dynamics in online communities of creation,” *Research Policy* (42:4), pp. 941–953.
Sievert, C., and Shirley, K. 2014. “LDAvis: A method for visualizing and interpreting topics,” in *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*, pp. 63–70.
Siggelkow, N. 2007. “Persuasion with case studies,” *Academy of Management Journal* (50:1), pp. 20–24.
Star, S. L. 1999. “The Ethnography of Infrastructure,” *American Behavioral Scientist* (43:3), pp. 377–391.
Star, S. L., and Ruhleder, K. 1996. “Steps toward an ecology of infrastructure: Design and access for large information spaces,” *Information systems research* (7:1), INFORMS, pp. 111–134.
Taylor, M. B. 2013. “Bitcoin and The Age of Bespoke Silicon How Bitcoin Works : User Perspective Bitcoin Mining : Miner’s Perspective,” in *Proceedings of 2013 International Conference on Compilers, Architecture and Synthesis for Embedded Systems (CASES)*, Montreal, Canada, pp. 1–10.
“The Bitcoin Hype And The Potential Disruptive Power Of Blockchain Technology.,” 2018. *Forbes.com* (available at https://www.forbes.com/sites/forbestechcouncil/2018/02/06/the-bitcoin-hype-and-the-potential-disruptive-power-of-blockchain-technology/#59ccd71d76c8).
Tilson, D., Lyytinen, K., and Sørensen, C. 2010. “Digital infrastructures: The missing IS research agenda,” *Information Systems Research* (21:4), pp. 748–759.
Tiwana, A., Konsynski, B., and Bush, A. a. 2010. “Platform evolution: Coevolution of platform architecture, governance, and environmental dynamics,” *Information Systems Research* (21:4), pp. 675–687.
Um, S., Yoo, Y., Wattal, S., Kulathinal, R. J., and Zhang, B. 2015. “The Generative Architecture of a Digital Ecosystem : A Network Perspective.,”
Vaast, E., and Walsham, G. 2009. “Trans-situated learning: supporting a network of practice with an information infrastructure,” *Information Systems Research* (20:4), pp. 547–564.
Vaast, E., and Walsham, G. 2011. “Grounded theorizing for electronically mediated social contexts,” *European Journal of Information Systems* (22:1), pp. 9–25.
Venkatesh, V., Brown, S. a., and Bala, H. 2013. “Bridging the qualitative-quantitative divide: Guidelines for conducting mixed methods research in Information Systems,” *Management Information Systems Quarterly* (37:3), pp. 855–879.
Wareham, J., Fox, P.B. and Cano Giner, J.L., 2014. Technology ecosystem governance. *Organization Science*, *25*(4), pp.1195-1215.
Weber, S. 2004. *The success of open source* (Vol. 368), Cambridge University Press.
Wenger, E. 2000. “Communities of Practice and Social Learning Systems,” *Organization* (7:2), pp. 225–246.
Wörner, D., Von Bomhard, T., Schreier, Y.-P., and Bilgeri, D. 2016. “The Bitcoin Ecosystem: Disruption beyond financial Services?,” *Twenty-Fourth European Conference on Information Systems (ECIS)*, pp. 1–16.
Yin, R. 2003. *Applications of Case Study Research (Applied Social Research Methods)* (4th editio.), Thousand Oaks: Sage .
Yoo, Y., Boland, R. J., Lyytinen, K., and Majchrzak, A. 2012. “Organizing for Innovation in the Digitized World,” *Organization Science* (23:5), pp. 13–32.
Yoo, Y., Lyytinen, K., and Yang, H. 2005. “The role of standards in innovation and diffusion of broadband mobile services: The case of South Korea,” *Journal of Strategic Information Systems* (14:3), pp. 323–353.
Zittrain, J. L. 2006. “The Generative Internet,” *Harvard Law Review* (119), pp. 1974–2040.